


Doctoral Dissertation
Doctoral Program in Computer and Control Engineering (33.th cycle)

Natural and multimodal interfaces for human-machine and human-robot interaction

Francesco De Pace
* * * * *

Supervisor
Prof. Andrea Sanna, Supervisor

Politecnico di Torino
TBA

I hereby declare that, the contents and organisation of this dissertation constitute my own original work and does not compromise in any way the rights of third parties, including those relating to the security of personal data.

. . . . . .

Francesco De Pace
Turin, TBA

Summary

Main goals of this Ph.D. dissertation are the design, the implementation and the validation of innovative natural interfaces able to efficiently and effectively support the user when interacting with different kind of machines and systems. The interfaces represent one of the most critical aspect of an interaction system. They act as contact points between the virtual world and the real one. Hence, their development must be carefully planned. In the first part of this thesis, the analysis of several Natural User Interfaces (NUIs) is presented, discussing their underlying mechanisms and highlighting their weaknesses and strengths. Then, among all the possible NUIs, this dissertation will focus on the use of Virtual and Augmented Reality (VR/AR) interfaces to improve the human-machine/human-robot interaction domain with particular interest for the Industry 4.0 context and serious gaming scenario. The VR and AR technologies will be firstly presented by analyzing their functioning and work flow. Afterwards, several original works regarding the use of AR and VR in the Industry 4.0 domain will be presented and detailed. Specifically, by analyzing how AR interfaces are currently employed to improve the efficiency of smart factories, some works related to the use of virtual interfaces to enhance maintenance and training operations will be detailed. Furthermore, virtual robotic teleoperation systems will be also considered, presenting some original works related to the use of RGB-D cameras and immersive VR interfaces to accurately control industrial robot arms. The AR and VR technologies will be also combined in the third chapter, discussing how hybrid virtual environments can be effectively developed, additionally analyzing the impact of the field-of-view on the usability of the virtual interfaces in the gaming context.

Acknowledgements

And I would like to thank my tutor Prof. Andrea Sanna and Dr. Federico Manuri for their continuous support and precious advises. Moreover, I would like to acknowledge the COMAU Italian company for its assistance and the Professors Mark Billinghurst and Minas Liarokapis for the opportunity to collaborate with the Empathic Computing Lab and the Department of Mechanical Engineering of Auckland, New Zealand.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

A User Interface (UI) can be defined as "*the medium through which the communication between users and computers takes place*" [181]. Given an input (normally from a human user), the machine computes the output which in turn is given back to the user as a feedback. The first rudimentary machines forced users to provide input using inefficient and complex systems (e.g., punched cards or paper tapes). Then, in 1968, Douglas Engelbart showed a combination of input/output interfaces using a new device of his own invention: the mouse [289]. This new input interface allowed users to provide input by simply moving a virtual cursor, displayed on a video interface. If until Engelbart the functionality of the input paradigm was not considered as fundamental as the machine itself, with the invention of the mouse it became clear that the input modality would have been increasingly important to properly interact with the machines. Despite it was possible to create a "perfect" machine, its performance would have been really limited by the users' input interface [99]. Hence, it became clear that the interaction between humans and machines was of primary importance to develop a stable and effective system. The very wide topic of the humans-machines communication is the main subject of the Human-Computer Interaction (HCI) science [181] and nowadays it is possible to find a plethora of different types of interfaces.

Figure 1.1 shows an high-level view of the main input/output interfaces, clustered by category. The input interfaces have been divided in four different branches: (i) Body Gestures, (ii) Voice, (iii) Brain, and (iv) Controller interfaces. Although this dissertation mainly focuses on the Natural and 3D User Interfaces, the Controller interface group has been added to the graph to provide readers a complete overview of the different types of input interfaces. However, this particular group will not be discussed in this work. Referring to Fig. 1.1, the top-right red rectangle highlights the so called Natural User Interfaces (NUIs). This peculiar set of interfaces encompasses those systems that allow the users to interact with the machines

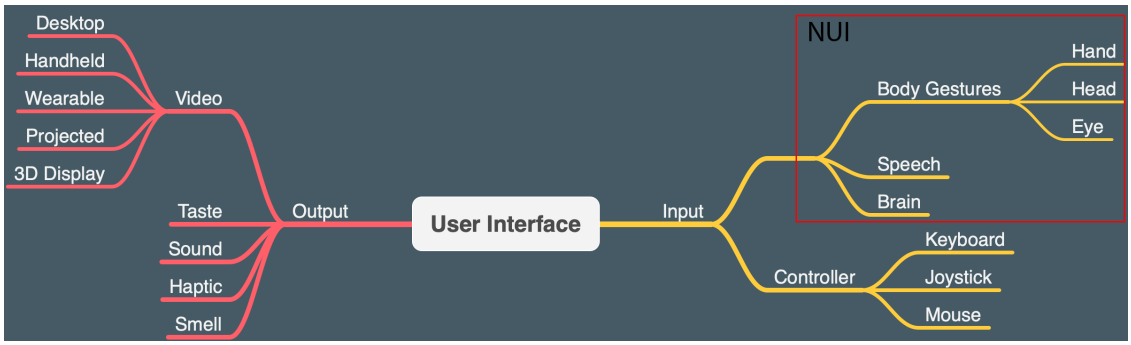


Figure 1.1: The Input and Output interfaces. The top-right rectangle highlights the so called Natural User Interfaces

without the necessity to learn the underlying interface mechanism[1]. Similarly to the input interfaces, the output ones have been divided in five distinct branches, corresponding to the five different human senses: (i) Video, (ii) Taste, (iii) Sound, (iv) Haptic and (v) Smell.

In the following sections, both input and output interfaces will be presented discussing their underlying mechanism.

1.1 The User Interfaces

1.1.1 The Natural Input Interfaces

Independently of the specific input interface, a NUI requires a tracking or recognition system. Since the underlying systems may greatly differ from one to another, the Body, Voice and Brain interfaces will be discussed separately in the following sections.

Body Gestures

Body gestures can be divided in three different categories: (i) Hand, (ii) Head and Eye gestures (Fig. 1.1). Hand gestures are normally classified into *static* and *dynamic* gestures [61]. The former accounts only for the position and orientation of the hand without considering any type of movement. On the other hand, the latter deals with the variations of the position and orientation of the hand with respect to time. In order to detect the hand gestures, it is possible to employ at least two different methodologies: the *contact based* and *vision based* approaches [361].

[1] Although in this work the above definition of NUI is employed, it should be noticed that there is currently an open debate around the word *natural* and its intrinsic meaning [275]

The contact based approaches rely on the use of physical sensors that should be worn or manually used by the users. These sensors may employ different tracking devices, such as mechanical [219], haptic [475], ultra-sonic [207] and inertial [384] sensors. Although it has been show that the contact based approaches can be quite effective in acquiring the hand's gesture data, they require a direct contact between the users and the sensors, making them bulky and uncomfortable to be used. Therefore the vision based approaches have greatly attracted the attention of the researchers, allowing to capture the hand's movements without forcing users to wear any type of device. These approaches normally employ RGB or RGB-D cameras and possibly markers (active or passive) placed on the user's hand. There are at least two different types of hand gesture representations: appereance and 3D model based [378]. Appearance representation methods try to create a 2D hand model using color [46], silhouette geometry [34], deformable gabarit [206] and motion [269] models. On the other hand, 3D model based methods try to create the 3D shape of the hand using 3D texture volumetric [280], 3D geometric [171] or 3D skeleton [218] models.

A hand gesture recognition system is composed by four different steps [378, 57]: (i) detection, (ii) gesture modeling, (iii) feature extraction, and (iv) classification. The detection step involves capturing the gestures' data using contact or vision based systems. Then, the acquired data have to be properly modeled depending on the application's type. One of the simplest methods to represent static gestures consists in using appearance approaches. However, since these approaches struggle in identifying complex static hand gestures, 3D static methods are usually employed. They are classified in *discriminative* and *generative* approaches [57]. The first ones do not create a hand 3D model but they employ classifiers trained to map unknown hand shape data with appearance features. Instead, generative approaches try to fit a 3D model of the human hand directly using the acquired data. Dynamic gestures can be modeled using motion information and they usually require the tracking of the human hand centroid. Once it has been tracked, its position, velocity and acceleration can be determined to create a model of the hand motion. One of the main issues is related to the detection of the so-called *gesture spotting*, the beginning and ending points of a specific gesture in a continuous motion. After the gesture has been properly modeled, features should be extracted to recognize the related gesture. Several descriptors are available to extract features, such as Fourier [164], discrete cosine transform (DCT) [5], wavelet [189], curvature [488] and histogram of gradients (HOG) [116] descriptors. Then, the collected features have to be recognized using a suitable classifier. The most commons are based on *k-Means* [120, 273] or *k-Nearest neighbour* [440] algorithms. Support vector machines and Hidden Markov model have been also successfully employed to recognize hand features [79, 237]. Finally, some more recent approaches employ artificial neural networks or deep networks to recognize the hand data [170, 231].

The detection of head movements and gestures is becoming increasingly important to find out humans' intentions. These gestures allow humans to "select" objects of interest or improve the human-computer communication [355]. Head movements can be recognized using several approaches [355]. Computer vision methods rely on the analysis of a single image or of a video sequence. Head pose information can be extracted from a single image using several approaches [302]: appearance template methods [30, 314], detector arrays [500], nonlinear regression methods [392, 501], manifold embedding methods [290, 415], and finally flexible [232, 71] or geometric [140, 185] models. Video analysis is effectively carried out using tracking approaches [302]. In [262], Scale-Invariant Feature Transform (SIFT) descriptors are employed to match features points among different video frames. Once detected, the relative angle is computed determining the global head pose. Kupetz et al. [236] have proposed a head pose estimation using infrared (IR) cameras and LEDs. An infrared LED array is positioned on the user's head and its movements are tracked using the method proposed in [122]. The detected movements are subsequently used to control an electric wheelchair. Several other computer vision approaches exists (e.g., Lucas-Kaskade algorithm [502], 3D models [490], etc.); for a complete and comprehensible review refer to [302, 355]. Sensor methods rely on the use of ad-hoc hardware (e.g., accelerometers, gyroscopes, etc.) to detect the head movements. Some examples can be found in [224], whose authors employ neural netowrks to calssify the accelerometer data or in [222], where gyroscope data are used to determine the global head pose. It is worth noticing that it is also possible to find commercial Virtual and Augmented Reality devices that detect and track the head pose. Some examples (but not limited to) are the HTC Vive[2], the Oculus Quest[3] and the Microsoft HoloLens 2[4]. Finally, it is possible to detect the head movements using acoustic-signal methods [379] that estimate the head direction by localizing the origin of the human voice.

Similarly to the head movements' recognition, eye detection has greatly captured the attention of the researchers. The eyes themselves and their movements can convey emotions, needs and aspirations [320], playing a key-role in the human-machine interaction context. Eye movements can be effectively tracked using computer vision methods. They usually comprehend four steps: image acquisition, eye detection, eye tracking and gaze estimation [163]. During the first step, an image containing the eye and its surroundings is acquired, then the position of the eye is detected in the second step. Once detected, the movements of the eye can be tracked in the third phase and eventually the gaze direction can be estimated in the last step. Several techniques exist to detect and track the eyes movements:

[2] https://www.vive.com/eu/

[3] https://www.oculus.com/quest/

[4] https://www.microsoft.com/it-it/hololens

pattern recognition approaches [360, 433], shape-based techniques [62, 228] and feature-based methods [213, 212]. Independently of the detection technique, the aforementioned approaches employ a camera that captures the eye and the region around it. Other approaches instead use IR cameras that illuminate the eye, localizing the corneal reflection [395]. Examples of IR systems can be found in [491, 492]. Finally, due to thier intrinsic capability of being non-intrusive, the computer vision methods have been widely researched and employed. However, it is worth mentioning that there exist alternative methods that employ sensors placed around the eye that analyze the electric potentail, known as electrooculogram [145].

Speech

Speech is one of the most important form of communication. It allows us to convey intentions, actions and, more importantly, it is a natural and effective way to exchange information among humans. Due to their importance, speech interfaces are expected to be employed to control machines and to exchange data with them [131]. A speech recognition system is usually composed of four different steps [142, 313]: (i) signal acquisition, (ii) pre-processing, (iii) feature extraction, and (iv) classification. During the first phase, the audio signal is acquired using dedicated hardware (e.g. microphones). Then, the signal is pre-processed, removing noise and dividing the signal itself into small frames that will be analyzed in the following step [203]. Once pre-processed, the frames are analyzed to extract meaningful features that will classified in the last step. There exist several approaches to extract features from audio frames, the more relevant being the following: Principal Component Analysyis [382, 394], Mel Frequency Cepstral Coefficients [375, 235] and Linear Predictive Coding [153, 160]. Finally, the extracted features are classified in the last step. It is possible to find several methods to effectively classify the audio features: acoustic phonetic approaches [405, 247], pattern recognition methods [354], support vector machine [98, 493] and artificial neural networks [95, 8] (for a complete classification, please refer to [131, 203, 313]).

Brain

Brain computer interfaces (BCIs) are increasingly employed to provide humans with the capability of controlling machines by analyzing the brain activities [2]. Normally, the human brain controls the muscular and skeleton systems which in turn allow us to interact and complete the desired action. A BCI instead allows human operators to *directly* complete the action without involving the muscular and skeleton systems [445]. A BCI is composed of four major steps: (i) signal acquisition, (ii) artifact processor, (iii) feature extraction, and (iv) classification [25, 445]. The first step involves the signal acquisition from the brain activities. It can be done with *invasive*, *partially invasive* or *non-invasive* techniques. The former

refers to read the brain signals by placing sensors inside the grey brain matter. On the other hand, partially invasive techniques place sensors outside the grey brain matter, reducing the risk of damage to the brain itself. Finally, the latter are the most used ones and they employ electrodes placed outside of the skull. Several non-intrusive techniques exist, the most well known are: Electroencephalogram [104], magnetic and functional magnetic resonance imaging [22, 408], Electrocorticography [356] and positron emission tomography [343]. Before acquiring the data, the signal is amplified easing the data acquisition. However, the acquisition may produce artifacts that are subsequnetly removed in the artifact processor step [25]. Then the brain signal is analyzed in the feature extraction step. Features can be extracted using several methods such as discrete Wavelet transform [396], fast Fourier transform [472] or Wavelet Packet Decomposition [444]. Finally, the features are examined to extract useful information that will be converted in the related user's action. Several classification methods are available, the most well-known (but not limited to) are: Support Vector Machine [422], Common Spatial Pattern [37], multi layer perceptron [220] and random forest methods [75]. Interested readers should refer to [445] for a comprehensive review of BCI.

1.1.2 Output Interfaces

Human beings use their senses to receive stimuli from the environment. Hence, the output interfaces have been divided according to the five human senses: (i) Smell, (ii) Taste, (iii) Sound, (iv) Haptic, and (v) Video (Fig. 1.1). Since the underlying working mechanism may greatly differ from one interface to another, in the following sections each of them will be separately presented and discussed.

Smell Interfaces

It is estimated that humans can recognize over a trillion of different fragrances [50]. Besides recognizing different odours, smell is also employed to define a spatial mapping of the environment [100, 191] and to track objects [126, 350]. Given the importance of such sense, the smell interfaces have increasingly captured the attention of researchers and therefore there exist several techniques to create the sensation of smell. A smell interface can be defined as an *olfactory display*, a device capable of "*being programmed to create an olfactory stimulus by emitting odorous molecules (chemo-stimulation) or creating a sense of smell (electro-stimulation)*" [336]. Such devices can be classified based on the their mechanism for producing smell and the most well-known are the following: ultrasonic atomization [38, 10], atomization through Venturi effect [215], evaporative diffusion [349, 177] and electro-stimulation [165, 414]. Olfactory displays have been used in several areas. Baus et al. [28] (Fig. 1.2) suggest that bad artificial smell can be more effective than pleasing odours in the gaming context. Military training has also benefited

from olfactory displays[5] as well as the food perception [132]. Furthermore, scents have been used in [38] to provide message notifications and in [495] to augment driving experiences.

Figure 1.2: A smell interface can foster a virtual reality experience. Figure published in [28], license courtesy provided by Springer Nature N. 5020801086828, Mar. 02, 2021.

Taste Interfaces

Our taste buds can perceive five different basic tastes: bitter, sweet, salty, umami, and sour [93]. They can be simulated using at least three distinct approaches: chemical, electrical and thermal simulation [73]. Through chemical components, users can experience different aromas by combining citrid acid (sour), sodium chloride (salty), monosodium glutamate (unami), caffeine (bitter), and glucose (sweet) [316]. Maynes-Aminzade [286] proposed an Edible User Interface that replaces the *painted bits* of a desktop monitor with tangible *edible bits*. Murer et al. [301] presented an interactive lollipop, called *LOLLio*, that acts as a haptic input device in a gaming context. The results show that sweet and sour tastes can greatly foster the game experience. Another example can be found in [94], which proposed a virtual reality headset that is capable of stimulating all five senses. Chemicals have been used to stimulate both taste and smell whilst touch, hearing and sight have been digitally stimulated. On the other hand, electrical stimulation is commonly achieved by placing electrodes in the oral cavity. Tongue papillae have been electrically stimulated in [345] by using a single silver electrode. The results show that the sour sensation has been easier to convey to the users with respect to salty and bitter sensations. Lawless et al. [246] analyzed the metallic taste generation by comparing electrical stimulation with metal-based stimulation and solutions composed of ferrous sulphate and divalent salts. Thermal stimulation is

[5] https://www.newscientist.com/article/dn8282-invention-soldiers-obeying-odours/

instead achieved by warming or cooling specif areas of the tongue. As an example, it has been demonstrated that it is possible to reproduce the sweet sensation by warming the front edge of the tongue whereas saltiness and sourness can be experienced by cooling the whole tongue [76]. A combination of electrical and thermal stimulations is proposed in [359]. The system is capable of stimulating three distinct sensations (sour, bitter and salty), with two different intensity levels (strong and mild). Finally, referring to commercially available solutions, Planet Licker[6] is one of the first commercial interface that employs taste as an input modality.

Sound Interfaces

From an high-view perspective, audio output interfaces have been divided in two macro-areas: 2D and 3D audio interfaces [368]. The former refers to systems that play an audio signal without any form of spatialization. On the other hand, the latter encompasses systems that can control and change audio parameters (e.g., intensity, direction, etc.), localizing the audio source spatial position. Regarding the 2D audio interfaces, Moreno et al. [296] proposed a video-audio hybrid system to improve the locomotion of blind people. By exploiting a mobile phone camera, doors and obstacle are detected and notified to the user by means of different sounds. The sound amplitude, frequency and envelope can be customized and tuned according to the user's needs. In [42] a 2D sound interface is employed to foster the evaluation of the shape of industrial products. When the users move their fingers over a haptic membrane placed over the product's surface, different sounds are played according to the absolute value, sign and discontinuities of the curvature. Similarly, Covarrubias et al. [74] mapped additional curve parameters (e.g., tangency, errors and discrepancies) to distinct sounds. They evaluated the system by comparing different typologies of sound and the results show that Modal Synthesis realistic complex noise sounds are the most effective ones to experience the shape curvature. However, the authors argued that the sounds should be carefully employed because the users can be distracted from the haptic and visual information. Referring to the 3D audio interfaces, an audio source can be virtually localized by means of binaural synthesis [205]. It exploits two different cues: the interaural time difference (ITD) and the interaural level difference (ILD) [450]. These cues are commonly modeled by head related transfer functions (HRTFs) that describe the transfer function from the sound source to the user's ears [481]. Early in the past, spatial audio interfaces have been successfully employed to improve the web browser navigation [150] or desktop applications [413]. More recently, 3D audio systems have been used in the robotic context for mobile robot navigation [187]. By using an array of microphones, a mobile robot can localize the sound sources

[6] http://a-o.in/games/pl/

elevation analyzing the ITD and ILD. Frauenberger et al. [127] proposed a 3D audio interface to help blind people in exploring virtual environments. By wearing headphones, the users can perceive the sound reflections and reverberations that help them to explore the environment. Spatial audio cues can be also conveyed using bone conduction. In [270], a stereo bone conduction system has been compared with headphones to evaluate its effectiveness in presenting spatial auditory stimuli. The results show that a stereo bone conduction system can be effectively employed as a spatial audio interface with the same extent of traditional headphones. Lock et al. [265] proposed a 3D bone conduction interface to guide people with visual impairments. In order to help the users to detect the target's elevation angle, the authors suggest to tune the audio signal pitch. The proposed approach is compared with a traditional bone conduction system and the results show that adjusting the tone's pitch can greatly improve the localization of the target's elevation.

Haptic Interfaces

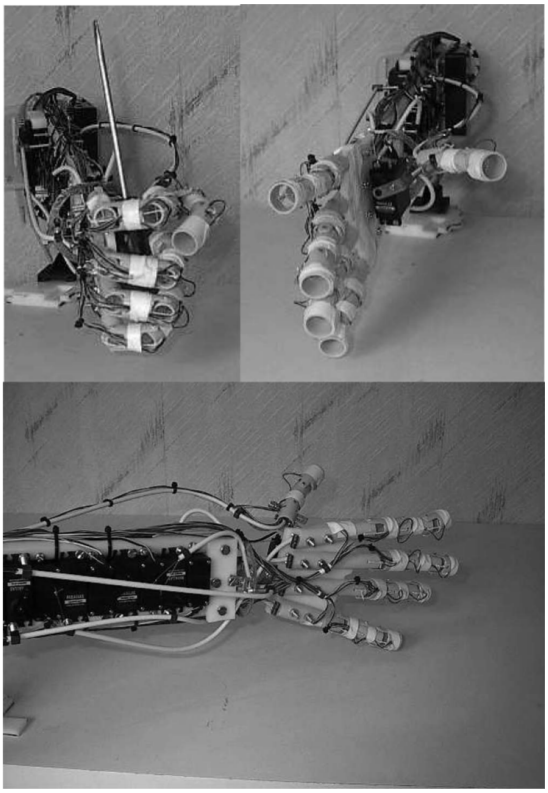


Figure 1.3: The robotic hand equipped with haptic sensors. Figure published in [202] ©[2011] IEEE.

Haptic interfaces can convey cutaneous signals providing the users with the sensation of "touch"[7]. The human brain processes these signals, giving us the *haptic impression* of an object [77]. Haptic interfaces have been employed in several domain, from robotic [202, 342] to tele-rehabilitation [291]. In [202], three distinct robotic hands are equipped with haptic sensors providing the capability of recognizing shape, texture and hardness properties of several objects (Fig. 1.3). Similarly, Petrovskaya et al. [342] proposed a custom Bayesian approach capable of efficiently compute the pose of a 6 degrees-of-freedom (DOF) robotic hand during grasp operations. Regarding the rehabilitation context, patients can carry out a series of tasks (moving a virtual ball through a virtual maze or squeezing a virtual cube to make it pass through a tiny hole) using a PHANTOM device [291]. Besides the robotic and rehabilitation context, haptic interfaces have been also used in [341] to provide the ability of "feeling" illustrations to children with visual impairment or in [89] to improve the game experience of a virtual billiards game. An emerging area where haptic interfaces are increasingly employed is represented by the virtual reality simulations. In such fictitious environments, since the commercial virtual reality devices are not capable of properly conveying haptic sensations [469], the haptic simulation is usually emulated and conveyed using ad-hoc devices. Culbertson et al. [77] analyzed three haptic features (surface friction, tapping transient and texture vibrations) and their influence on the touch sensation for virtual objects. The results show that the improvements to the touch sensation are directly related to the intensity of the complementary surface's property (slipperiness, hardness, or roughness). Quadcopters are emloyed in [184] to convey haptic feedback in a virtual environment. Moving around in the real environment, the quadcopters can be positioned in the locations of the virtual objects, acting as haptic feedback proxy. The results demonstrate that the proposed approach greatly improves the sense of presence with respect to the vibrotactile controllers.

Video Interfaces

Desktop

Video interfaces have been divided in four different categories: (i) Desktop, (ii) Handheld, (iii) Projected, and (iv) Wearable. Regarding the Desktop interfaces, the first computer monitors used the cathode ray tubes (CRTs) technology, discovered by Julius Plucker and Johann Wilhelm Hittorf [281]. Then, thanks to technological advancements, CRT monitors have been replaced by Liquid Crystal Displays (LCDs) that take up less space, provide lighter image and consume less power. Desktop interfaces are nowadays become the de-facto standard to carry out computer tasks and they are employed in many different areas. First studies related

[7]The adjective *haptic* derives from the Greek word *haptesthai*, that means *to touch* [452]

to the evaluation of the effectiveness of monitor displays show that desktop interfaces (along with mouse) are statistically superior to menu selection interfaces with function keys [362]. Other preliminary studies evaluated the possibility of desktop extension using handheld devices [303] or whether the monitor interfaces are left or right hand biased [272]. Desktop interfaces have been also employed to visualize 360-degree videos [41]. The main results suggest that spatial understanding can be increased using visual boundaries and it is not related to the correct perception of directions. Toma et al. [447] compared a monitor interface with an immersive virtual reality (VR) system in a 3D-Computer-Aided Design (CAD) assembly scenario. The results show that, although the VR system allowed to complete the tasks in less time with respect to the desktop one, the immersive interface required a higher physical workload. A similar study is proposed in [186] and the results suggest that the users could not clearly perceive the objects' dimensions using the desktop interface. Finally, González et al. [149] compared wearable Augmented Reality (AR) interfaces with AR desktop ones. Since their findings show that wearable interfaces are more error prone than the desktop ones, the authors argued that at this state of the art there are no reasons to employ a wearable AR interface instead of a traditional desktop one for authoring tools.

A handheld device is defined as *[...] an object that it can be held and used easily with one or two hands*[8] and thus an output video handhed interface can be defined as *an interface that displays information using a handheld device*. Early research focused on determining which features should be implemented to develop a reliable and effective handheld interface [148] or whether the user interfaces developed for a specific device could be used with different devices (desktop and handheld) [204]. Then, thanks to technological advancements, the handheld devices have been equipped with sensors (e.g., accelerometers, gyroscopes, magnetometers etc.) that allow to compute the device pose providing innovative interaction modalities. Hachet et al. [161] presented one of the first 3-DOF interface based on target recognition and video analysis. Their results show that this type of handheld interface can greatly improve the manipulation of large 3D models. The ability of manipulating 3D models has been explored also in AR scenarios. In [346], a comparison between an AR handheld interface and an interface based on fixed annotations for inspection tasks is proposed (Fig. 1.4). The main results show that the AR handheld interface allowed the users to complete the tasks with less errors and workload than the non-AR one. Tanikawa et al. [434] compared three different hand held mid-air gesture interfaces for virtual object manipulation tasks. According to their results, an interface that allows the users to visualize a virtual rod (going from the device to the virtual object) provides the highest success rate. Other examples of handheld interfaces for virtual objects manipulation can

[8] https://dictionary.cambridge.org/dictionary/english/handheld

be found in [376], where authors compared device perspective with user perspective rendering or in [154], which analyzes hybrid interaction techniques (touch gestures and device movements) for handheld devices. Finally, multi-user environments are also considered in [496], showing that collaborative handheld interfaces can greatly improve the game play experience of ordinary tabletop games.

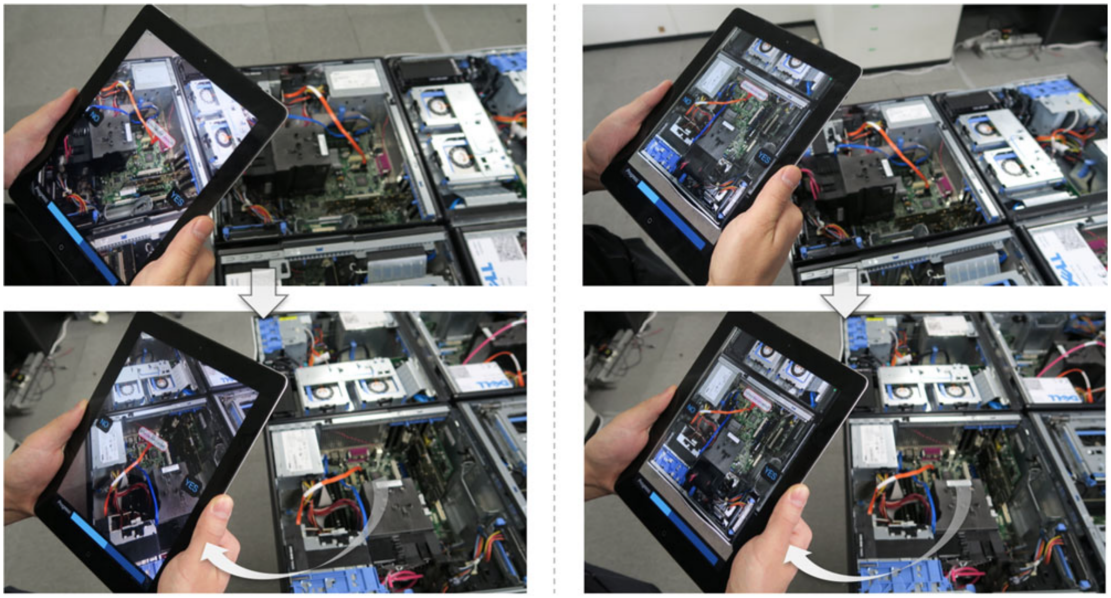


Figure 1.4: Left-column: the AR handheld interface. Right-column: the interface with fixed annotations. Figure published in [346] ©[2018] IEEE.

Video interfaces can be also displayed using wearable devices. Such devices are commonly referred to the term *Head-Mounted Displays* (HMDs) and they can be worn on the head displaying information using monocular or binocular optics. Their application domain encompasses different areas (e.g, engineering, architecture, medicine, etc. [399]) and they are commonly employed along with VR and AR technologies. As an example, in the medicine context, Ortega et al. [325] employed an HMD that displays data in front of the users' dominant eye, allowing them to visualize fluoroscopy images during orthopaedic operations. The results show that the HMD reduced the total number of times the doctors moved their attention from the operative area. Similarly, the HMDs have been employed in [192] to provide patients with the ability of visualizing their own sonography images in real-time or in [179] to help visual impaired users to detect objects by adjusting the brightness of the screen according to the distance from the objects. Moving from the medicine context, the HMDs have been also used in the industry domain. Zheng et al. [503] evaluated the effects of display positions on the quality of maintenance processes. The results show that data displayed with central eye-wearable devices led to lower task times than peripheral eye-wearable ones. The robotic context has increasingly

attracted the attention of the researchers for several purposes, such as control [326], [157] and teleoperation [258] using wearable devices. Oyekan et al. [328] used HMDs to evaluate human reactions in human-robot collaborative environments. Similarly, a VR HMD is employed in [339] for training purposes allowing the users to interact with a virtual robot positioned in a robotic cell reconstructed with depth sensors (Fig. 1.5). Other interesting works can be found in [176], where an HMD is used to visualize and approve the robot path or in [103] for obstacle avoidance purposes. Another application domain is represented by the education and learning context. A comparison among HMDs, desktop and CAVE systems is carried out in [6] to evaluate the effects of different interfaces on the learning process. The main findings suggest that HMDs greatly improved the learning outcomes with respect to the other two conditions. Reiners et al. [364] found out that the sense of presence can be increased by wearing HMDs in standing up positions, whereas it immediately decreases in sitting down configurations. Beside the learning context, several other HMDs features (e.g., field-of view, duration of the exposure, etc.) have been analyzed to determine whether they affect the cybersickness symptoms [7]. The results show that HMDs with fixed interpupillary distance may increase the cybersickness symptoms, lowering the overall user experience. Finally, in addition to the video and see-through AR HMDs, some researchers are investigating the effectiveness of the so called *retinal head-mounted display* that can overcome the accomodation and convergence problems of the conventional HMDs [199, 254].

Figure 1.5: The operator can practice with the robotic cell using the VR immersive device. Figure published in [339], license courtesy provided by Springer Nature N. 5020810787182, Mar. 02, 2021.

The projected interfaces have been divided according to the physical dimensions of the projector: (i) mobile-size, (ii) medium-size, (iii) room-size, and (iv) full-dome. Mobile-size projectors are *hand-sized* projectors that can be used in mobile conditions. Kim et al. [223] proposed a handheld projector that can overlay virtual information on real objects recognized by a tiny camera. The users can interact

with the real objects by using a pre-defined set of actions (e.g., flicking, tilting, etc.) and, depending on the action, different virtual information is displayed close to the real object. Similarly, a simulated mobile projector system is presented in [412]. By integrating the system with a digital pen, the projector can augment the paper documents, providing additional information. Furthermore, the system supports multi-user interaction to improve the collaboration and data management. The second category encompasses projectors that are usually employed in fixed positions, hung over walls or ceilings. They have been used in several domains, such as the robotic [459] or medical [487] contexts. Vogel et al. [459] employed a fixed projector system to display the operative area of a high-payload industrial robot. Depending on the hazard level, the operative area is visualized by three different colors (Fig. 1.6).

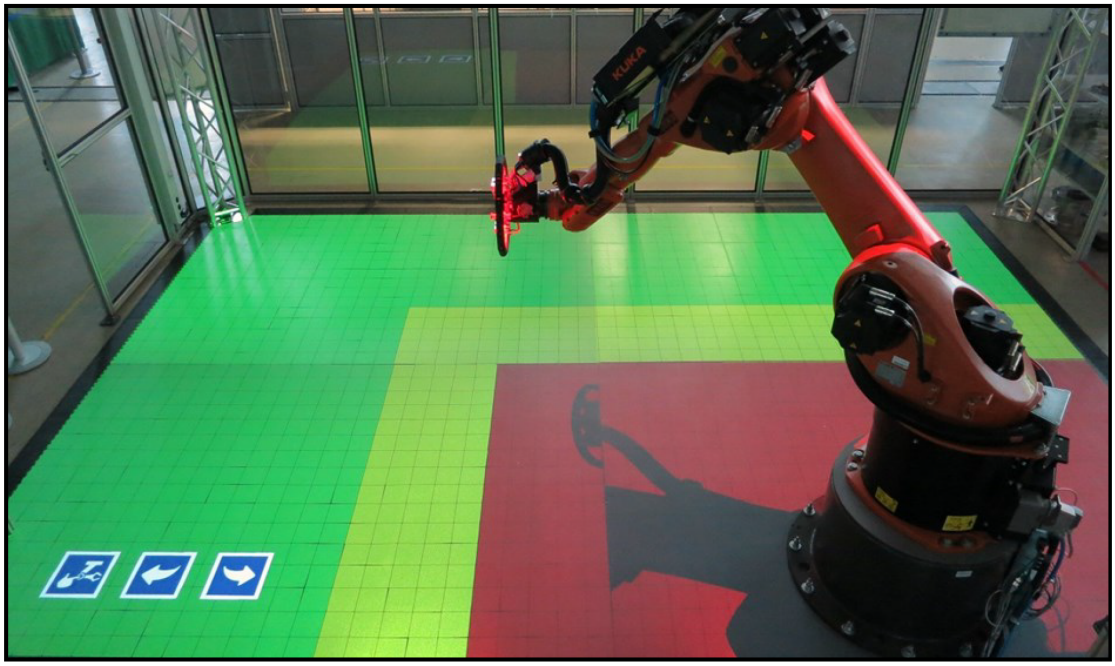


Figure 1.6: The three different colors used to highlight the robot operative area. Figure published in [459] ©[2016] IEEE.

In the medical context, a robot arm has been combined with a medium-size projector to improve cranio-facial surgery operations [487]. The patient's head is firstly scanned and a 3D model of the skull is derived. Then, several surgery data (e.g., intersection and osteotomy lines, bore holes, etc.) are directly superimposed on the patient's head with an accuracy of ±1mm. Other examples of use of medium-size projectors in the medical context can be found in [476, 477]. Differently from the previous categories, room-size and full-dome projectors are usually employed to improve the sense of presence and realism of the virtual environments. Examples

regarding the use of room-size projectors can be found in [256], where pilots can test new aircraft design in an aerospace simulator or in [51], where the authors evaluated the effectiveness of a gesture interface (refer to [300] for a complete review concerning room-size projectors and virtual environments). Finally, full-dome projectors can display virtual information on very wide areas, providing multi-user experience. Chastenay [59] evaluated the effectiveness of such systems in teaching astronomical phenomena. By projecting astronomical data on a planetarium ceiling, the users can visualize the lunar phases with both allocentric and a geocentric perspectives. The results show that the full-dome system greatly improved the understanding of the astronomical phenomena. Similarly, the effectiveness in teaching basic Earth science concept has been tested in [420] and the results suggest that the the combination of different instruction modalities with large projected displays can foster the learning experience.

3D displays are systems capable of displaying 3D images without forcing users to wear any particular device. They can be classified according to the underlying technology [180]: (i) volumetric, (ii) swept-volume surfaces, (iii) holography, (iv) optophoretic, (v) plasmonic, and (vi) lenticular lenslets. Hirayama et al. [180] proposed a levitating volumetric display capable of acoustically trapping particles that will be subsequently illuminated with red, green and blue lights. High-viscosity liquid microbubble voxels are generated in [234] by means of femtosecond laser pulses. The microbubble voxels' colors can be changed by controlling the illumination of the light sources. Swept-volume surfaces are characterized by a rotating panel that generates a display volume. A large swept-volume display capable of visualizing full-motion 3D video is presented in [381]. Light-emitting diodes are controlled by a field-programmable gate array positioned on the rotating panel image panes. Similarly, a swept-volume display in presented in [243] along with a static volume display that uses transparent crystals as a projection volume. Light diffraction is employed by holographic displays to generate 3D images. Blanche et al. [35] developed a holographic display composed of a single 4x4in^2 photorefractive polymer. One interesting property of the photorefractive polymers is that they can record and project new three-dimensional images every few minutes, allowing users to watch holograms for several hours. Other examples of holographic displays can be found in [3] and [23]. Regarding optophoretic displays, Smalley et al. [410] proposed a 3D display based on photophoretic optical trapping techniques. Cellulose particles trapped with spherical and astigmatic aberrations are scanned with red, green and blue light generating 3D images. An example of plasmonic display is presented in [317] where physical matter, excited by high-intensity laser, emits light from arbitrary three-dimensional positions. Finally, Gao et al. [135] proposed a study of lenticular lenslet displays. The results show that Fresnel-lens-array with eccentric pupil can improve the brightness and viewing field of the generated 3D images.

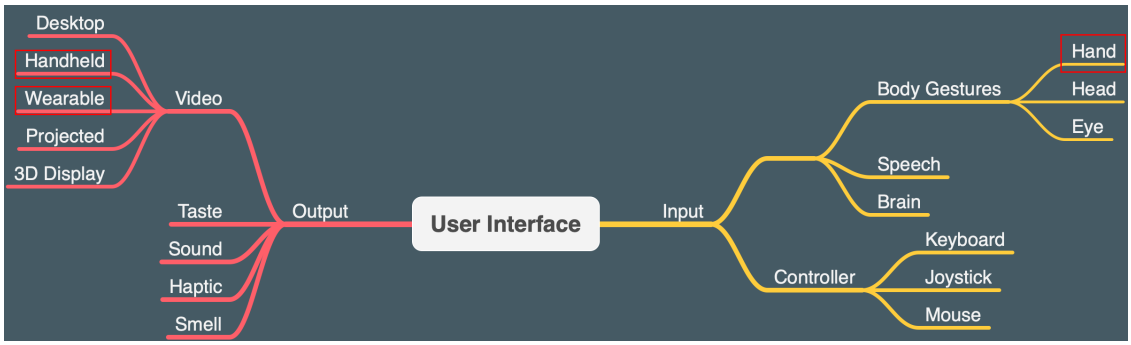


Figure 1.7: The red rectangles highlight the interfaces considered in this dissertation.

1.1.3 Conclusions

The study and development of UIs for the HCI context is indeed a very wide topic that requires both technical and user-centred skills. Since the aforementioned UI classification encompasses a great number of different interfaces, it has been decided to narrow down the number of interfaces to be researched and improved for this Ph.D. dissertation. Regarding the NUI, this dissertation will focus on those interfaces that mainly employ hand gestures recognition systems (contact and optical based). For the output interfaces, this thesis will focus on the video interfaces with particular interest for the wearable and handheld ones used in VR and AR environments (Fig. 1.7).

In the following section, the VR and AR interfaces will be deeply discussed, with particular emphasis on the immersive VR and video AR interfaces.

1.2 Virtual and Augmented Reality Interfaces

Virtual and Augmented Reality interfaces are strongly linked and they present both common and different characteristics. As Milgram and Kishino explain in [295], both interfaces are part of the same *reality-virtuality continuum* (Fig. 1.8): a VR environment consists of a "*pure*" virtual scenario, entirely fictitious and completely detached from the real environment. On the other hand, an AR interface presents digital contents *in* the real world, that is, the real world is *augmented* by virtual assets in real-time and users can interact with both the environment and the digital assets at the same time. Despite VR and AR interfaces are part of the same continuum, they present some differences and peculiarities (e.g., interaction strategies, visualization techniques, etc.) that require an ad-hoc and specific discussion.

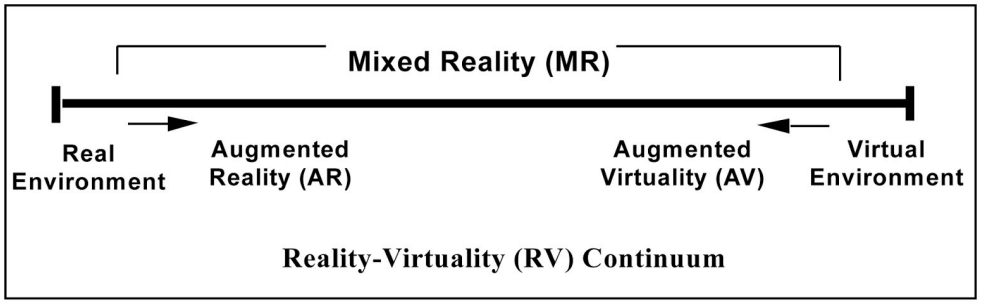


Figure 1.8: The reality-virtuality continuum.

1.2.1 VR

VR interfaces allow users being *physically present in a non-physical world* [129]. HMDs are among the most used devices to *immerse* users in virtual worlds. The VR devices are equipped with sensors and ad-hoc hardware that greatly help reducing the gap between the real environment and the virtual world [128]. The first HMD patent dates back to 1916 when Albert B. Pratt invented a head-based periscope display, basically a "*gun adapted to be mounted on and fired from the head of the marksman*" [351]. Then, in 1929, Edward Link presented the first flight simulator[9], a mechanical system that allowed pilots to be trained without being on a real plane. From then, many discoveries have followed (e.g., Sensorama in 1962 [173], Sketchpad in 1964 [427], the *ultimate display* in 1965 [425], etc.), but probably one of the most relevant event in the VR history happened when Sutherland presented in 1968 a tracked stereoscopic head-mounted display [426]. A cathode ray tube display, equipped with two separated optics, that was capable of presenting distinct images to each human eye, thus creating the illusion of virtual immersion. Since then, the VR devices have been greatly improved and nowadays it is possible to find on the market several HMDs. Although these devices present differences in terms of features and tracking mechanisms, the main underlying technology is shared among all the devices and it will be discussed in the next section.

HMD Technologies

In its most basic form, an HMD is composed by a couple of displays that project separated images to each eye. They exploit the so called *parallax phenomena*, the intrinsic capability of the human eyes to perceive depth by visualizing the same image from two slightly different perspectives. Although the parallax allows users to perceive depth, it is not sufficient to provide a fully immersive experience. When

[9] https://web.archive.org/web/20120317171710/http://library.binghamton.edu/specialcollections/findingaids/linkcoll_m3.html

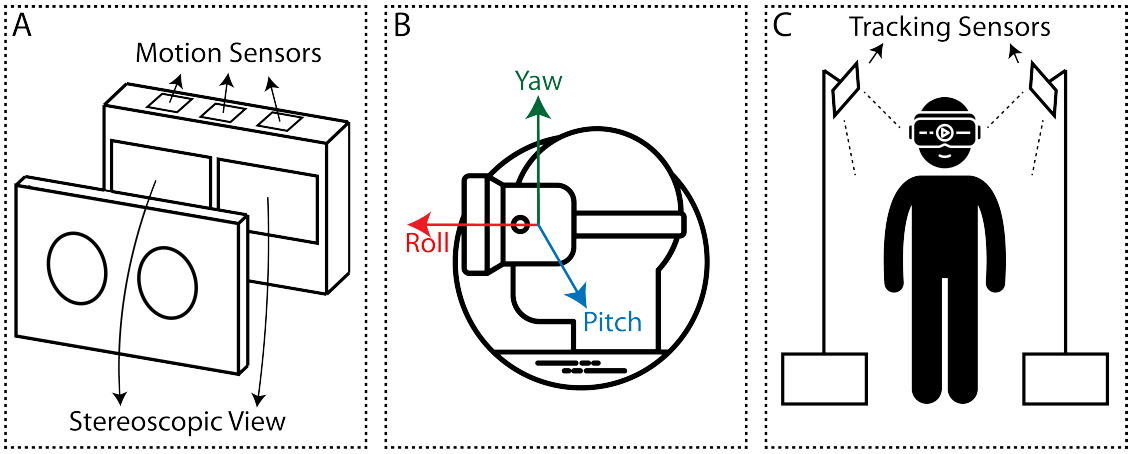


Figure 1.9: A) the stereoscopic view. B) the motion sensors used to compute the head orientation. C) the external tracking sensors used to compute the user's position.

the user is wearing an HMD, he/she can change the head orientation and thus he/she expects to perceive a change in the orientation of the virtual environment. To do so, at least a 3-DOF tracking system should be employed to compute the modification in the head orientation. However, to achieve a fully immersive experience, a 6-DOF tracking is required, thus considering also the head position. The head orientation is usually tracked using motion tracking sensors (gyroscopes, accelerometers and magnetometers) whereas the head position is computed using external tracking system, such as structured or IR light systems. Figure[10] 1.9 shows an illustrative example of a wearable VR system. The stereoscopic displays provide separated images to the human eyes, creating the perception of depth. The motion sensors are used to compute the head orientation (in terms of roll, pitch and yaw), allowing users to change the view of the virtual environment according to the head orientation. Finally, the position of the head can be computed by using external sensors that track markers positioned on the HMD surface.

Nowadays, the HMDs are usually coupled with controllers that allow users to interact with virtual objects. Depending on the user interaction, it is possible to employ several typologies of controller, ranging from simple "traditional console" joysticks to 6-DOF controllers (Fig. 1.10 shows some of the most common VR controllers). Console joysticks probably provide users with the most traditional form of interaction: the users can "virtually" move by using the thumbsticks (see Sec. 1.2.1 for additional information regarding the locomotion in VR environments) and they can interact by pressing the joystick's buttons. On the other hand, 6-DOF

[10]Virtual reality headset by Vectors Market, human by Gan Khoon Lay and vr by supalerk laipawat from the Noun Project.

Figure 1.10: Some well-known VR controllers. From left to right: The Microsoft Odyssey, the Oculus Touch, the PlayStation VR and the HTC Vive controllers.

controllers allow the users a more natural interaction than the console ones. Their position and orientation is usually tracked using the same approach adopted for the HMDs and thus the users can virtually "touch" the virtual assets, without pressing any button on the controller. Furthermore, some most recent controllers are also equipped with additional sensors that allow to compute the fingers' position on the controller itself, that is, the controller is aware which buttons the user is touching. This information can be particularly useful to improve the sense of immersion, allowing to animate in real-time the virtual hands according to the position of the real user's fingers.

Tracking

Since the tracking methodology plays a key role in an immersive VR system, a brief discussion related to the different approach is proposed in this section. The device that communicates its position and/or orientation to a central control unit is called *position sensor* [397]. The general configuration usually comprehends several position sensor units attached to the object being tracked and at least one sensor unit placed at a known position. There exist several tracking methodologies, each of them with strengths and weaknesses that impose limitations to their usage. Some of the most well-known tracking methods that can be found in a VR system are the following: (i) electromagnetic, (ii) mechanical, (iii) optical, (iv) ultrasonic, (v) inertial, and (vi) neural [397]. An electromagnetic tracking system employs a pair of *transmitter-receiver* units. Three orthogonal coils placed inside the transmitter

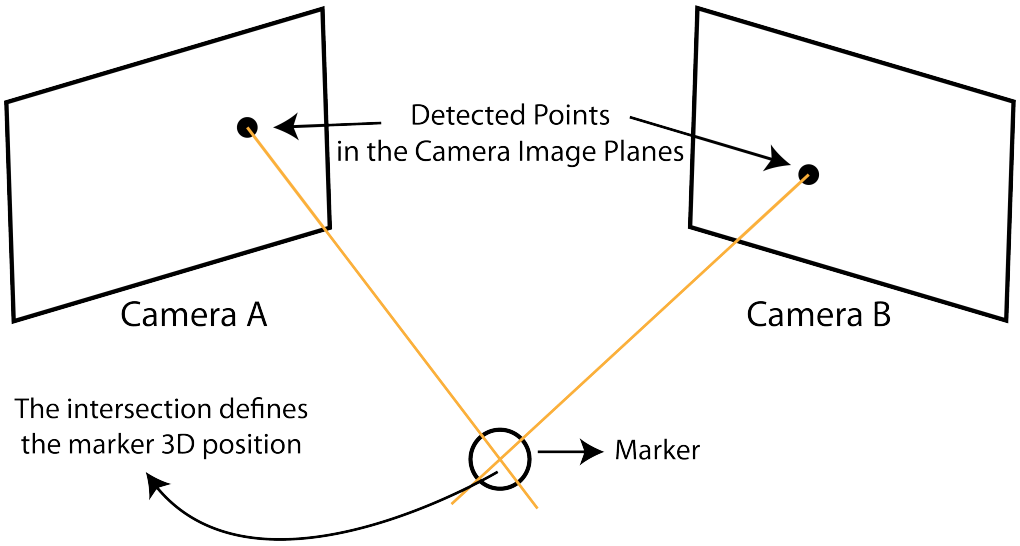


Figure 1.11: Starting from the respective cameras, the virtual rays can uniquely identify the 3D marker position.

create a magnetic field that in turns generates currents in the other coils positioned in the receiver worn by the user. The receiver analyzes the signal and it computes the position and orientation of each coil with respect to the fixed unit (i.e., the transmitter). One of the major advantage of the electromagnetic system is that there are no line of sight limits. However, metal objects can interfere with the electromagnetic field and it has a very limited operative range (3-8 feet). Mechanical tracking systems use mechanical booms to compute the user's position and orientation. A mechanical boom can be seen as a kinematic chain whose terminal part can be worn or grasped by the user. When the user moves around the terminal part, the positions of the chain joints are computed determining the pose of the user. One of the main drawback of such systems is that the operative space is limited by the boom's extension. Optical tracking systems use cameras to track the position and orientation of the position sensors. Although a camera can work in the visible spectrum tracking paper markers, most of the systems employ IR cameras and markers. At least a couple of cameras are required to uniquely identify the IR marker's position and orientation (Fig. 1.11).

High-pitch sounds emitted at fixed intervals are employed by ultrasonic systems to determine the distance between the transmitter and the receiver. The advantage of these systems is that they employ common hardware, such as microphones and speakers, but noisy environments can really limit their performance. Moreover, to accurately determine the transmitter-receiver distance, there must be no obstacles between the source and the receiver. Changing in acceleration, inclination and gyroscopic forces are instead measured by inertial tracking systems to compute the sensors' relative motion [121]. Typical sensors are accelerometers, gyroscopes and

inclinometers. These tiny sensors are positioned on the object being tracked and they are connected to a central control unit that filters and elaborates the sensors' data. Even if accelerators and gyroscopes can be theoretically employed to compute a 6-DOF pose, they accumulate errors over time producing inaccurate measurements. Hence, they are normally employed to compute only the orientation of the tracked object. The accelerometer data are also usually combined with magnetometer ones, in order to mitigate the accuracy degradation over time. Finally, neural tracking systems track individual body parts instead of the user's pose. Small sensors are directly attached to the skin, measuring muscle contractions and/or nerve signal changes. By analyzing the collected data, it is possible to determine which movement the user is making and with which part of the body.

VR Locomotion

Locomotion in virtual environments is extremely important to provide a full immersive experience. It is expected that users can move their virtual body in the same (or similar) way they walk and run in the real world. However, the locomotion presents several challenges that should be overcome to achieve an immersive and safe experience. With *safe* it is meant an experience that does not generate negative symptoms, the so called *cybersickness* [111, 137, 455]. Cybersickness can be defined as a *"psychophysical response to the exposure of perceptual illusions in a VR"* [455]: several characteristics of the VR environment (e.g., display frame rate, latency, etc.) may generate negative symptoms and the locomotion strategy should be carefully planned to reduce the negative effects of cybersickness [69, 287]. Beside doing their best to mitigate the cybersickness symtomps, the locomotion techniques try also to provide users the ability to move beyond the physical boundaries imposed by the specific tracking approach. In other words, these techniques may allow the users to walk for kilometers whilst being physically restricted to an area of a few meters. The research is very active on this particular topic and several locomotion strategies can be adopted. The most well-known are the following: real-walking, walking-in-place, controller, gesture-based, teleportation, redirect walking, arm swinging, head-directed and chair-based [39]. Real walking techniques allow users to freely walk in a limited area with a 1:1 match between the real and the virtual displacements. Examples of real walking techniques can be found in [388], where the authors proposed a mechanical stilt to walk up and down steps, or in [372], where real walking navigation is achieved by using foot-mounted inertial sensors. When using walking-in-place approaches, the users move in the virtual environment by walking in place, that is, by doing step-by-step movements. This locomotion type can be achieved by means of treadmill or stepping-like devices (see for example the Virtual Sphere project [409] or the Stepper Machine [44]). Controller techniques allow the users to move by using the joystick buttons and/or

the thumbsticks. The joystick typology can vary from simple controller [304] to keyboard [16] and trackball [44]. If gesture recognition is supported, gesture-based systems provide users with the ability to give a movement command that is translated in a virtual motion [53, 118]. Due to its simplicity and effectiveness, teleportation is one of the most employed approaches. By pointing towards the final destination, the user is immediately teleported there, allowing to cover great distances. The pointing action can be done by controllers [489] and/or gestures [45]. By exploiting redirect-walking methods, the users can freely walk in a real limited area while being capable of moving in an unlimited virtual space. Special techniques are used to create a subtle mismatch between the virtual and real worlds, without the user noticing it. As an example, Nescher et al. [309] proposed a probabilistic algorithm to maximize the walking action whereas in [498] a novel approach is presented to predict the user's locomotion path. Arm swinging techniques provide users with the ability to move in the VR world by swing their real arms while being stationary. External cameras can be used to track the arm movements [242] or alternatively tracked controllers [484]. Finally, with the head-directed methods, the HMD movements are directly translated into virtual displacements [53, 423] whereas with the chair-based approaches, the users sit on a chair and its rotation and tilt are used to control the virtual avatar in the VR environment [226].

1.2.2 AR

With respect to VR, "almost" at the other side of the reality-virtuality continuum (see Fig. 1.8) it is possible to find augmented reality. "Almost" because the true opposite of a pure VR environment is the real one whereas the AR scenarios *augment* the real world by digital contents. AR interfaces have increasingly attracted the attention of researchers for their intrinsic capability to be connected to the real environment. Human beings put naturally attention toward the real world rather than to the virtual one. Despite pure VR interfaces (but also common digital devices such as smartphones and personal computers) provide access to an incredible amount of data, the digital contents are usually completely detached from the real world, creating a *rupture* between the real and virtual environments. This is where AR interfaces come in: they allow users to visualize and interact with both environments at the same time providing a natural bridge between the real and virtual contents.

AR dates back to the '60s when Sutherland anticipated its definition with a little known but very important sentence: "*[. . .]The user of one of today's visual displays can easily make solid objects transparent — he can "see through matter!*" [425]. After few years, his dream of a display capable of seeing through matter came true and he presented the first AR HMD, the so called "Sword of Damocles" [426]. It was a device equipped with head tracking and optical see-through that allowed to visualize digital contents overlaid on the real environment. Then, from the '70s to

the '80s, several researches have experimented with systems that allowed to interact with digital contents superimposed on the real environment. But we have to wait until the 1992 to read for the first time the term *augmented reality* on a research paper [443]. The authors proposed a wearable AR system that provided airplane workers with the ability to visualize wire bundle assembly schematics directly overlaid on the real aircraft. One year later, Fitzmaurice [119] presented the first AR handheld device. It was a tethered LCD screen tracked by means of a magnetic tracking system that was capable of displaying spatial information correctly aligned with the real world. At that time, the proposed AR systems were limited to a single user interaction, there was not support for multiuser environments. To overcome this limitation, Schmalstieg et al. [387] proposed Studierstube, the first multiuser collaborative AR system. By wearing a tracked HMD, multiple users could visualize the same digital assets, seen from different and individual perspectives. One year later, Feiner et al. [114] presented the first outdoor AR system. The users could visualize digital assets correctly aligned with the outdoor environment by using a see-through HMD tracked with GPS and motion sensors. This system was used for several applications, but probably it became famous with the first outdoor AR game, *ARQuake* [441]. This game was a port of the very well known first-person shooter game Quake and it allowed users to shoot virtual zombies positioned and rendered in the real environment. Until 1999 there was no standard or globally recognized solution to develop AR applications and the AR development was strictly limited to the research labs. This situation changed when Kato and Billinghurst released ARToolkit in 1999 [211]. It consisted in the first set of open-source tools to develop AR applications by using printed markers. Since then, the development of AR applications has increasingly grown and nowadays it is possible to find several commercial toolkits and devices, such as the Vuforia Software Development Kit (SDK)[11] or the Microsoft HoloLens 2[12]. As for VR, although there are several commercially available AR devices, the main underlying technology and visualization pipeline is commonly shared among the different solutions that will be deeply discussed in the following sections.

AR Technology

In 1997 Azuma listed the three main characteristics of an AR system [20]:

- combines real and virtual;
- interactive in real time;
- registered in 3D.

[11] https://www.ptc.com/en/products/vuforia/vuforia-engine

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

The first interesting consequence of such definition is that Azuma did not specify the output device. Although an AR system can be developed using a great variety of output devices and it is not limited to the video interfaces (e.g., AR systems can be developed using olfactory, sound, etc output interfaces, see Sec. 1.1.2), this dissertation will focus only on the video AR interfaces. The first characteristic ("combines real and virtual") is quite straightforward: virtual and real elements should be combined to create a new augmented environment, where both kind of information exist. The second one implies the concepts of *real time* and *interactivity*. If the former strictly depends on the single use case, the latter requires that humans and machines are strongly linked together. While the user is moving in the AR environment, the system continuously tracks his/her movement, computing the user's pose. This information is finally used to correctly register and align the virtual assets with the real world (Fig. 1.12).

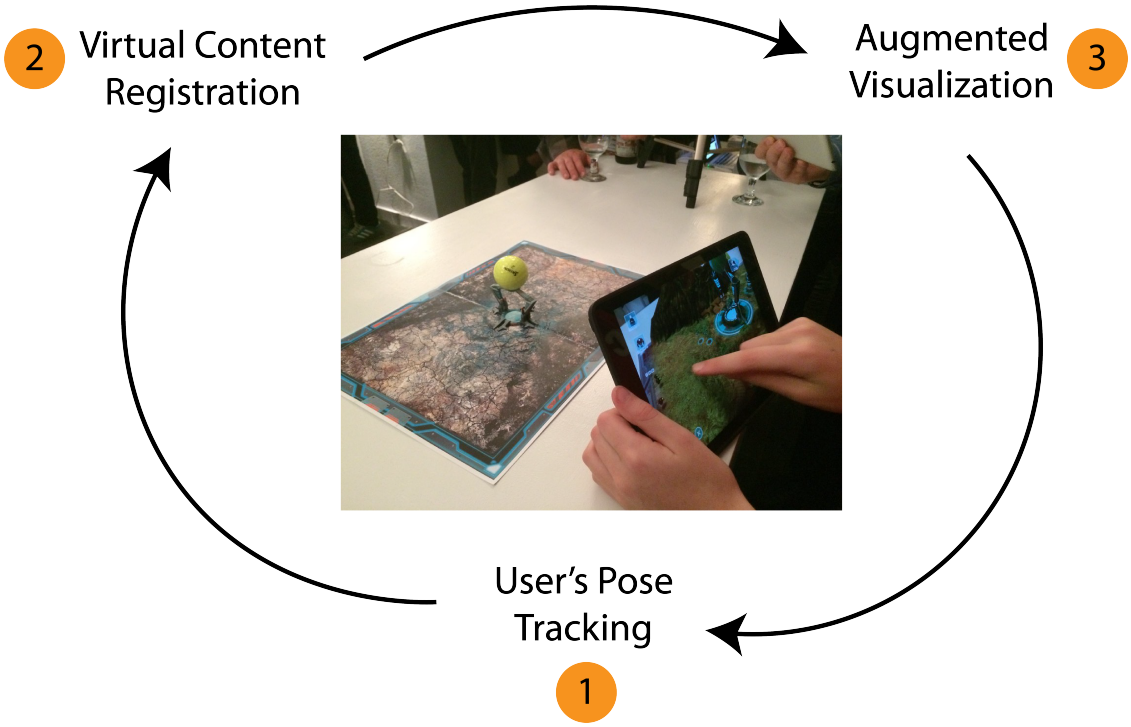


Figure 1.12: 1) The AR system continuosly tracks the user's pose. 2) The tracking data are used to correctly align the virtual contents to the real world. 3) The user can visualize the augmented scene.

An AR system is composed of at least three main components: (i) a tracking, (ii) a scene-generator and (iii) a visualization component. The tracking component is responsible for determining the camera's position and orientation. As for VR, it is possible to employ several distinct tracking techniques (see Sec. 1.2.1). However, since the vast majority of the video AR systems relies on the use of a camera, the

most employed tracking technique results to be the optical one. Optical tracking can be divided in marker-based and markerless tracking. With the first one, the camera, working in the visible spectrum, recognizes pre-defined markers positioned in the real environment. A marker recognition pipeline is composed of at least four steps:

- S1: image acquisition;
- S2: marker detection;
- S3: pose estimation;
- S4: rendering of the virtual assets.

During the first step, an image is acquired using a camera with a known corresponding mathematical model. Normally, a camera is modeled as a *pinhole camera* (Fig. 1.13), that describes the perspective projection of a world 3D point q to an image plane 2D point p. The perspective projection can be expressed in homogeneous

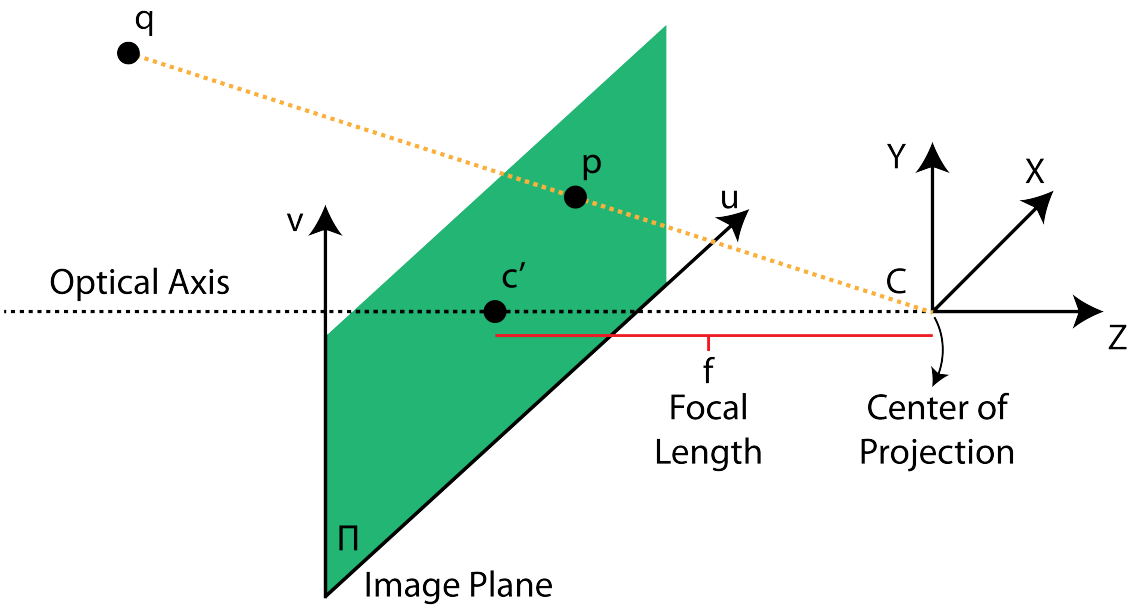


Figure 1.13: The pinhole camera. A virtual ray (in yellow), starting from the center of projection C, projects the 3D world point q on the image plane Π.

coordinates using a 3x4 matrix M:

$$\begin{bmatrix} p_u \\ p_v \\ 1 \end{bmatrix} = M * \begin{bmatrix} q_x \\ q_y \\ q_z \\ 1 \end{bmatrix}$$

where p_u, p_v are the coordinates of the point p lying on the image plane Π. M is a 11 DOFs matrix composed of *intrinsic* and *extrinsic* parameters. The former describe the camera internal parameter and they are represented by a 3x3 matrix K:

$$K = \begin{bmatrix} f_u & s & c_u \\ 0 & f_v & c_v \\ 0 & 0 & 1 \end{bmatrix}$$

where f_u, f_v represent the focal length (usually $f_u = f_v$), c_u, c_v describe the coordinates of the principal point c and s is the skew factor, different from zero only when the axis u and v are not perpendicular. The intrinsic parameters are usually computed during a calibration phase and they are treated as constants during the overall application cycle (unless the focal length can change, such as during zooming actions). The extrinsic parameters describe the position and orientation of the camera with respect to a known reference system. They are represented by a 3x4 matrix $[R|t]$ which in turn can be decomposed in a 3x3 rotation matrix R and a translation vector t. The main objective of the S2 and S3 steps is to compute the 3x4 $[R|t]$ matrix parameters. Detecting a marker means to analyze the image to extrapolate regions of pre-defined shapes (Fig. 1.14 shows several well-known markers and their shape) and to compare the detected regions with a pre-defined image (the marker itself). A marker is commonly represented as a black square surrounding a 2D barcode. The barcode can be uniquely identified and it expresses an unique orientation that defines the orientation of the marker itself. The detection starts with the camera image converted to a binary image (black and white). Dynamic thresholding is normally applied to compute the threshold value (see for example the Otsu's method [306]) or [327]. Then, the binary image is scanned for closed contours. Finding closed contours means searching for shapes that have a sufficient size (with respect to the original marker) and such that we can can fit a quadrilateral to the contour [467]. One popular approach to detect closed contours can be found in [428]. Once the closed contour is detected, the orientation of the marker is computed by sampling four points (the four corners) that will be used to compute the camera pose. Since the coordinate of a point q lying on a plane Π' (the marker itself) can be expressed with homogeneous coordinates, the mapping between q and the point $p \in \Pi$ can be modeled using a 3x3 *homography* matrix H. H can be determined using direct linear transformation [329] and singular value decomposition. Once determined, H can be used then to recover the $[R|t]$ matrix, used to compute the camera's pose (for the details, please refer to [386]). Finally, the virtual assets can be correctly rendered (in terms of position and orientation) in the output image using the camera's pose in the last S4 step.

Since the release of the popular framework ARToolkit [211], the research has been very active in this particular topic and it is possible to find several research

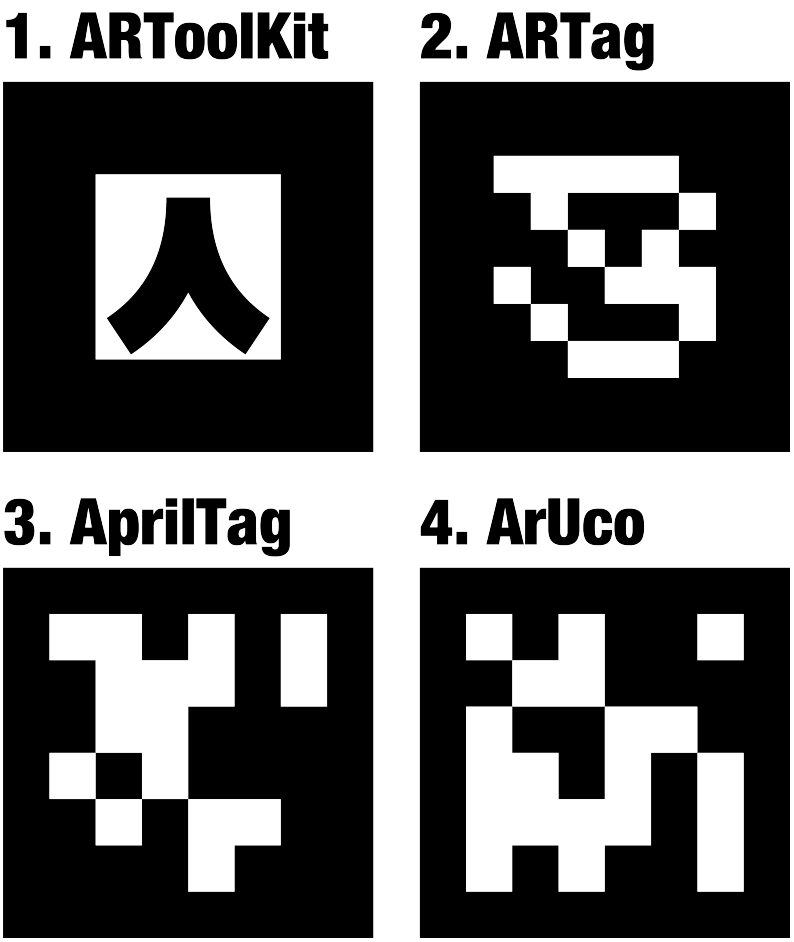


Figure 1.14: An example of different fiducial markers. Image provided by Cmglee - Own work, CC BY-SA 4.0.

papers regarding how to improve the marker recognition [306, 419] as well as commercial solutions, such as Vuforia[13] or VisionLib[14] SDKs. However, markerless techniques are increasingly attracting the attention of the researchers due to their capability to align the virtual assets without using artificial markers. Markerless approaches rely on the detection and tracking of natural features. During the years, several algorithms have been developed to properly identify *reliable* features. Reliable because a stable features should be robust to lightning changes, perspective transformations as well as rotation and scale. Some well known features detectors are the Harris Corners detector [168], which computes horizontal and vertical gradients to detect corners or the FAST detector [371], which is optimized for real-time

[13] https://developer.vuforia.com/

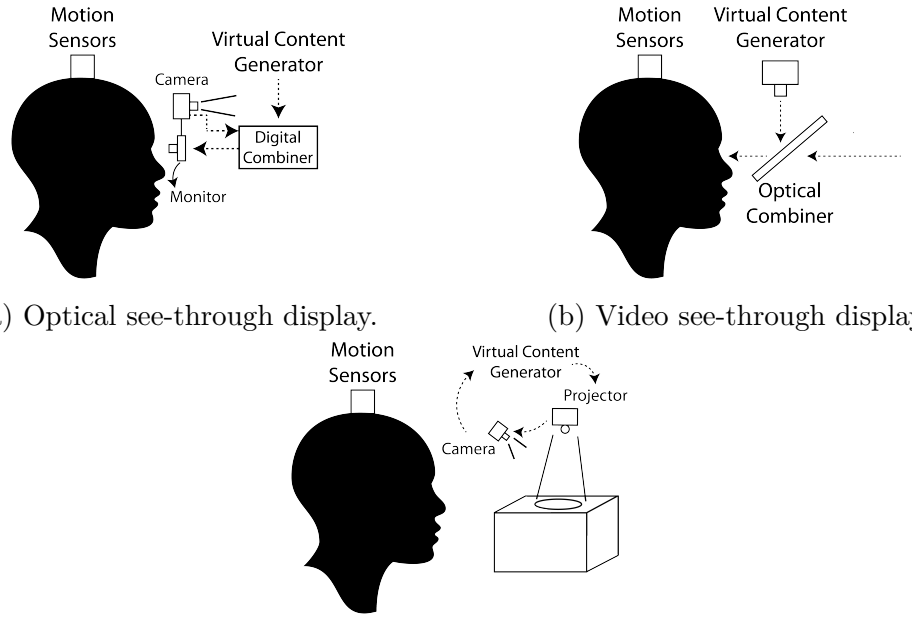
[14] https://visionlib.com/

applications. Once the features have been properly detected, they should be mathematically described in order to compute a data structure suitable for mapping the features of the current frame to the features of the other frames (see for example the SIFT descriptor [268]). Finding and tracking natural features is a well-known methodology, exploited by several tracking algorithms, such as the visual odometry [312] or PTAM [227] approaches. However, visual odometry suffers from drift over time, whereas PTAM works only with a sparse point cloud and it suffers from poor texture areas [386]. Modern approaches are capable of tracking a great number of points and are much more robust against poor tracking conditions. Recent technological improvements have made quite popular the use of RGB-D sensors to effectively adopt dense SLAM techniques. The KinectFusion algorithm [311] is one of the most representative examples of such techniques. It computes the camera's pose by minimizing the error in the alignment of the depth image of the current frame to previous one by using an iterative closest point (ICP) algorithm [17]. The alignment is iteratively applied until the error is deemed small enough or the number of iterations has reached a predefined threshold [344].

Independently of the tracking modality, once the camera's pose is computed, it is sent to the scene-generator module to correctly align the virtual assets with respect to the real environment. Hence, the virtual assets can be finally visualized in the last step, the visualization one that will be introduced in the next section.

Visualization - The Video AR Displays

How virtual assets are visualized plays a key-role in an AR system. Depending on the application requirements, different strategies can be adopted. There are two main display categories: *see-through* and *spatial* displays. See-through devices combine the virtual and real contents using optical lenses through which the users can view the augmented environment. They are further divided in *video* and *optical* see-through displays. Video see-through devices electronically combine the virtual assets with the real environment. Initially, a camera is usually employed to capture the real world and the related video stream is sent to the graphic processor. Thereafter, the virtual assets and the camera image are combined by firstly copying the video image into the frame buffer, that is, the video image is treated as a background image. Then, the digital contents (that have been previously aligned during the tracking phase) are simply added to the frame buffer, creating a combined image. Finally, the resulting image is shown to the user through standard monitors and/or displays (e.g., desktop, handheld, etc.). On the other hand, optical see-through devices rely on the use of optical elements that are either transmissive and reflective. An effective example is represented by a half-silvered mirror that lets the real light pass through so that the real world can be seen by the users. At the same time, the mirror reflects the virtual assets generated by a display positioned on the side of the mirror itself (or overhead). Hence, the virtual images appear as overlaid



(a) Optical see-through display.

(b) Video see-through display.

(c) Projected display.

Figure 1.15: An illustrative example of the different video AR displays.

over the real environment. Finally, the last category is represented by the spatial displays. These devices allow to superimpose the virtual assets directly on the real environment, without forcing the users to wear any kind of devices. A camera is usually employed to compute the camera's pose. Then, a projector displays the virtual assets, overlaid on the real objects. Figure[15] 1.15 shows the three different AR visualization types.

1.2.3 Conclusions

In this chapter, the VR and AR interfaces have been deeply detailed and explained, with particular emphasis on the immersive VR and the video AR interfaces. These specific interfaces have been widely researched and used in several domains. As an example, Zhang et al. [499] analyzed the effectiveness of an immersive VR interface in the mining context. Similarly, in [485] the authors showed that VR HMDs can be effectively employed to detect faults in 3D CAD models. The AR interfaces have been successfully used in the industry domain as well. See for example [141] where an AR interface is used to check discrepancies between a real object and its corresponding CAD model or in [383] where the authors employed

[15]Profile face by Sergi Delgado from the Noun Project.

an AR handheld interface to visualize underground infrastructures. Besides the industry context, the virtual interfaces have been also used in the medical [297, 307], autonomous vehicle [1, 92], cultural heritage [162, 333], tourism [264, 466] domains (and many others). Given the considerable number of domains, it has been decided to narrow down the application areas to two main categories, which are studied in this dissertation, the Industry 4.0 and the gaming context: they will be introduced and discussed in the next chapters.

1.3 Motivation and Overview of the Projects

HCI is a peculiar discipline as it is essentially composed of two distinct entities that are apparently in contrast with each other: the *human beings* and the *machines*. The human beings are characterized by high flexibility and adaptability and thus they can efficiently handle unforeseen scenarios or situations. On the contrary, the machines are devices that are composed of physical (the hardware) and digital (the software) layers and they essentially execute ordered instructions (the sequence of bits) with relative low levels of adaptability (at least in their most basic form). The HCI discipline researches innovative and efficient methodologies that can effectively bring together those two entities and one of its most powerful approach is represented by the so called *Human-centred design*. The ISO 9241-210:2019 [197] clearly details the Human-centred design main purpose: "Human-centred design is an approach to interactive systems development that aims to make systems usable and useful by focusing on the users, their needs and requirements, and by applying human factors/ergonomics, and usability knowledge and techniques. This approach enhances effectiveness and efficiency, improves human well-being, user satisfaction, accessibility and sustainability; and counteracts possible adverse effects of use on human health, safety and performance."

In order to develop software "*[. . .] usable and useful by focusing on the users, their needs and requirements, and by applying human factors/ergonomics, and usability knowledge and techniques*", the development itself should be constantly integrated and updated by user studies, thus allowing developers and researchers to effectively combine the human being and the machines. As previously introduced, the virtual interfaces are key-technologies of the HCI field and they are strongly related with the human beings. For instance, both AR and immersive VR technologies rely on the constant capture of the user's data (i.e., the tracking mechanism) to correctly display the digital contents (see Sec. 1.2.1 and Sec. 1.2.2). From military immersive VR experiences to cultural heritage AR applications, *the human being is always the final user of the virtual technologies*. One interesting consequence of this assumption is that the techniques and methodologies employed to assess the systems from a user-centred perspective can be easily moved from one context to another without loosing efficacy. Considering the two macro-categories discussed

in this Ph.D. dissertation, the Industry 4.0 and the gaming area, both have similar aspects when addressed using the virtual interfaces. As an example, both areas require systems and software that guarantee high levels of interaction and accuracy (e.g., to control a virtual character in a gaming scenario or to program an industrial robot arm in the Industry 4.0 context). Moreover, whether the users are following an AR procedure to assemble a robotic hand or they are playing against each other in a VR environment, the virtual interfaces should provide high levels of usability and low physical and mental workload scores to guarantee a satisfactory experience.

Moving from these considerations, this Ph.D. dissertation will discuss several original works related to the Industry 4.0 and gaming scenarios. Specifically, the use of AR and VR technologies in the Industry 4.0 context will be discussed in Section 2. Starting from an analysis of the different uses of the AR interfaces in the Industry 4.0 domain and in the collaborative robotics (Section 2.1 and Section 2.2), two different AR interfaces to visualize industrial robot faults will be presented in Section 2.3 and Section 2.4. Then, a collaborative mixed reality system to support the human operators during robotic training procedures will be discussed in Section 2.5. Finally, an immersive VR interface to remotely control an industrial manipulator will be presented in Section 2.6. The gaming context will be analyzed in Section 3 and Section 3.1 will present an evaluation of the usability of the AR and VR interfaces for competitive tabletop games. A similar assessment procedure will be discussed in Section 3.2 to evaluate the impact of the field-of-view on VR-AR first person shooter games. Section 3.3 will present a modular framework to ease the development of VR-AR environments. Finally, Section 4 will detail the final conclusions, discussing the overall results and limitations of this Ph.D. dissertation.

1.4 Specific Tools employed for this Thesis

The hardware and software tools used for the works described in this Ph.D. dissertation are presented in this section.

1.4.1 Hardware

Besides common handheld devices (e.g., smart-phones or tablets), two AR and three VR devices have been employed for the projects described in this thesis.

Epson Moverio BT-200

The Epson Moverio BT-200[16] are AR wearable smartglasses developed by Epson (Fig. 1.16). They employ two optical see-through displays that allow the users to

[16] https://tinyurl.com/6fz4wv4

Figure 1.16: The Epson Moverio BT-200 smartglasses along with their touch-enabled controller.

visualize digital contents placed in the real environment. They run the Android Operative System (OS) and the user can interact with the digital contents using an external touch-enabled controller. Table 1.1 shows the Epson Moverio BT-200 main specifications.

Type	AR			
Tethered	No			
OS	Android 4.0.4			
Weight (HMD)	88g			
Display	see-through	0.42 inch wide panel (16:9)		
Sensors	1 IMU	GPS	1 VGA camera	1 microphone
Processors	TI OMAP 4460 1.2Ghz Dual Core			
Memory	1GB RAM			
Storage	8GB			
Connectivity	IEEE 802.11b/g/n WiFi Miracast	Bluetooth 3.0		
Audio	No			
Ports	USB2.0			
FPS	60 fps			
Field-of-view	23°			
Gesture Recognition	No			
Speech Recognition	No			
Spatial Sound	No			
Controller	Yes (requried)	touch-enabled		

Table 1.1: The Epson Moverio BT-200 specifications.

Microsoft HoloLens (1st generation)

Figure 1.17: The Microsoft HoloLens smartglasses.

The Microsoft HoloLens[17] (1st generation) glasses are a wearable AR device capable of displaying virtual contents in the real environment (Fig. 1.17). With respect to the Moverio BT-200, the HoloLens glasses provide a higher resolution display and a larger field-of-view. They run a custom Microsoft Windows Mixed Reality OS and they support speech and gesture recognition. Table 1.2 shows the Microsoft HoloLens main specifications.

Oculus Rift DK 2 Kit

Figure 1.18: The Oculus Rift DK 2 Kit along with its IR camera.

The Oculus Rift DK 2 Kit[18] is an immersive VR HMD developed by Oculus VR (Fig. 1.18). It provides a 1920x1080 display (960x1080 per eye) which guarantees the visualization of high quality digital contents. The HMD position is tracked by means of an external IR camera. The camera tracks several IR markers positioned in the front part of the HMD. The rotation is instead captured by an internal IMU unit. Table 1.3 shows the Oculus Rift DK 2 Kit main specifications.

Type	AR	
Tethered	No	
OS	Windows 10 Windows Mixed Reality	
Weight (HMD)	579 g	
Display	2.3 megapixel see-through holographic lenses	2 HD 16:9 displays
Sensors	1 IMU, 4 understanding cameras, 1 depth camera, 1 2MP HD video camera, 1 microphone array	
Processors	Intel 32-bit (1GHz)	Microsoft Holographic Processing Unit
Memory	2GB RAM	
Storage	64GB (flash memory)	
Connectivity	Wi-Fi 802.11ac	Bluetooth 4.1 LE
Audio	3D audio speakers	3.5mm audio jack
Ports	Micro USB 2.0	
FPS	60 fps	
Field-of-view	30°	
Gesture Recognition	Yes	
Speech Recognition	Yes	
Spatial Sound	Yes	
Controller	Clicker (not required)	

Table 1.2: The Microsoft HoloLens (1st generation) specifications.

Figure 1.19: The Oculus Rift along with its controllers.

Oculus Rift

The Oculus Rift[19] is an immersive VR HMD developed by Oculus VR (Fig. 1.19). It provides a 2160×1200 display (1080x1200 per eye) which guarantees the visualization of higher quality digital assets with respect to the Oculus Rift DK 2 Kit.

[17] https://docs.microsoft.com/en-us/hololens/hololens1-hardware

[18] https://xinreality.com/wiki/Oculus_Rift_DK2

Type	VR
Tethered	Yes
Weight (HMD)	440g
Display	5.7 inch OLED 1920 x 1080 960 x 1080 per eye
Sensors	1 IMU
Connectivity	HDMI
Audio	No
Ports	USB
Refresh	60-75 Hz
Field-of-view	100°
Controller	No
Tracking	6-DoF
Tracking volume	72°H x 52°V
Positional tracking	External IR camera

Table 1.3: The Oculus Rift DK 2 Kit specifications.

The HMD position is tracked by means of two external IR cameras whereas the rotation is captured by an internal IMU unit. It is possible to buy two 6-DoF controllers that allow the users to interact with the virtual contents. The controllers are tracked by the same IR cameras used to compute the position of the HMD. Table 1.4 shows the Oculus Rift DK 2 Kit main specifications.

HTC Vive Pro

The HTC Vive Pro[20] is an immersive VR HMD developed by HTC (Fig. 1.20). The overall display resolution is 2880×1600, whereas each display provides a resolution of 1440x1600. Similar to the other VR devices, the HMD position is tracked by means of two external IR cameras, whereas the rotation is captured by an internal IMU unit. The minimum tracking area (using two IR cameras) is about 5m x 5m and it can be extended up to 10m x 10m using 4 IR cameras. Table 1.5 shows the HTC Vive Pro main specifications.

1.4.2 Software

The integrated development environment and the 3D computer graphics software tools used to develop the projects presented in this Ph.D. dissertation are

[19] https://www.oculus.com/rift/

[20] https://www.vive.com/us/product/vive-pro-full-kit/

Type	VR
Tethered	Yes
Weight (HMD)	470g
Display	2160×1200 (1080×1200 per eye)
Sensors	1 IMU
Connectivity	HDMI 1.3
Audio	Yes
Ports	2 USB 3.0 1 USB 2.0
Refresh	90 Hz
Field-of-view	110°
Controller	Yes
Tracking	6-DoF
Tracking volume	room scale: a 9ft x 9ft
Positional tracking	External IR cameras

Table 1.4: The Oculus Rift specifications.

Figure 1.20: The HTC Vive Pro along with the controllers and the two IR cameras.

presented in this section. Furthermore, the main external libraries and SDKs are detailed and discussed.

Unity

Unity[21] is a versatile cross-platform game engine. It has been used to develop all the projects proposed in this Ph.D. dissertation. The game engine provides support

[21] https://unity.com/

Type	VR
Tethered	Yes
Weight (HMD)	802g
Display	2880 x 1600 (1440 x 1600 per eye)
Sensors	1 IMU, IPD sensors, eye-tracking
Connectivity	Display Port 1.2 HDMI Bluetooth
Audio	Yes
Ports	USB-C
Refresh	90 Hz
Field-of-view	110°
Controller	Yes
Tracking	6-DoF
Tracking volume	room scale: minimum 5m x 5m
Positional tracking	External IR cameras

Table 1.5: The HTC Vive Pro specifications.

for Windows and macOS and, recently, it seems to have extended its support to the Ubuntu distribution[22]. Unity supports the C# programming language and it allows the developers to build applications for several platforms and OS[23]. The main functionalities of Unity can be easily extended by integrating external libraries and SDKs.

Blender

Blender[24] is a free open-source 3D software tool used to model and animate virtual objects and environments. It fully supports Windows, macOS and Linux and its main functionalities can be extended using the Python language. Referring to this Ph.D. dissertation, Blender has been employed to create the virtual assets (along with their animations) used for the developed VR and AR applications.

[22] https://itsfoss.com/unity-editor-linux/

[23] https://docs.unity3d.com/Manual/system-requirements.html

[24] https://www.blender.org/

Robot Operating System (ROS)

The Robot Operating System[25] (ROS) is a versatile open-source framework that greatly simplifies the development of robotic software. It natively supports the Ubuntu distribution although it seems to have recently extended its support to the Windows OS[26]. The ROS environment can be seen as a collection of *nodes* managed by a central node called *Master*. The nodes are small computational units that exchange data among each other through dedicated channels (the so called *topics*). The nodes can be developed using Python or the C++ language. There are essentially three different types of nodes: (i) a *publisher* node sends messages over a specific topic, (ii) a *subscriber* node listens for incoming messages on a specific topic and (iii) a *service* node advertises general services over the ROS network. More information regarding ROS can be found at[27].

Additional libraries and SDKs

Several additional libraries have been used to develop the AR and VR projects:

- the SteamVR unity plugin[28] has been used to simplify the development of immersive VR environments;
- the Microsoft Mixed Reality Toolkit (v1 and v2) has been used to interface with devices that support the Universal Windows Platform (e.g., the Microsoft HoloLens);
- the Vuforia SDK[29] has been used to simplify the creation of AR environments, allowing to easily detect image targets;
- the Ros#[30] library has been used to link a ROS network running on a Ubuntu distribution with a Unity environment running on Windows.

[25] https://www.ros.org/

[26] https://wiki.ros.org/Installation/Windows

[27] https://wiki.ros.org/ROS/Tutorials

[28] https://valvesoftware.github.io/steamvr_unity_plugin/

[29] https://www.ptc.com/en/products/vuforia/vuforia-engine

[30] https://github.com/siemens/ros-sharp/wiki

Chapter 2

Virtual Interfaces in Industry

Part of the works described in this chapter has also been published in [18, 82, 84, 83, 85, 278, 87]. An additional work presented in Sec. 2.6 has been accepted to be published at the ICRA 2021 conference[1]

So far, our society went through three different industrial revolutions. The first one happened at the end of the 18^{th} century when the society moved from an agriculture-based society to a mechanical one. Almost a century later, the discovery of the electricity gave rise to the 2^{nd} industrial revolution, providing innovative inventions, such as the telegraph and the telephone. Then, during the second half of the 20^{th} century, the third industrial revolution started, characterized by the discovery and improvements of electronics, telecommunications and computers. If the three mentioned revolutions have been characterized by mechanization, electrical energy and widespread digitalization [245], respectively, nowadays we are moving towards a fourth industrial revolution, where factories are expected to be completely autonomous and intelligent, the so called *Industry 4.0*. The term *Industry 4.0* was introduced by the German government to describe a high-tech strategy for future manufacturing industries [182]. The concept of Smart Manufacturing is at the core of this new revolution [208], meaning that factories are expected to become *smart* factories, helped by the adoption of new technologies, such as the Internet of Things and the Cyber Physical Systems [101]. The Industry 4.0 is characterized by nine main *pillars* [101] that stand for the nine main technologies that the factories are promoting and using to improve all areas of the production processes. Augmented Reality has been identified as one of the main technologies that can effectively support the fourth revolution. It has been used in several industry domains [82] and thus it is considered one of the main Industry 4.0 pillars. Another key-pillar is represented by the industrial robots that are expected to foster and improve the productivity of the forthcoming factories. Traditionally, the industrial robots work

[1] https://www.ieee-icra.org/index.aspx

in well-defined areas, completely separated by the human operators to reduce the risk of hazards. However, the new virtual interfaces provide innovative forms of interaction that can guarantee the safety of the human operators, thus allowing to remotely control the industrial robots using immersive VR head-up displays.

In the following sections, several works carried out for this Ph.D. dissertation are presented and discussed. Specifically, it will be discussed the use of virtual interfaces in the Industry 4.0 domain with particular interest for the collaborative robotics. Furthermore, some works regarding the use of AR and VR to visualize industrial robot faults and to improve the operators' training will be presented and detailed. Finally, new forms of virtual telerobotic systems will be detailed, with particular emphasis for the virtual robotic cells reconstructed with RGB-D sensors.

2.1 Augmented Reality in Industry 4.0

The role of AR in the Industry domain is relevant since it fosters both the product design and development. It helps identifying and avoiding design faults and errors during the early stages of the production process and it lowers the amount of physical prototype objects saving valuable cost and time. Moreover, this technology is deemed fundamental to improve and accelerate the development of products and processes in at least five industrial domains: maintenance-assembly-repair, training, products inspection, building monitoring, and in the Human-Robot Collaboration (HRC) context. In the maintenance-assembly-repair domain, the AR interfaces can be easily adopted to improve the productivity by reducing task and operative time. During training procedures, the AR technology results to be a powerful solution to improve the human operators' skills. The production inspection processes can benefit from the augmented visualization that highlights the item discrepancies. In the building monitoring operations, the AR interfaces provide the ability to detect building errors and deviations in a very intuitive manner. Finally, in the HRC context, the AR technology provides innovative interaction paradigms aiming at improving the collaboration between humans and robots.

In the next sections, each domain will be detailed and discussed.

2.1.1 Maintenance, Assembly and Repair

Since cost reduction is one of the main goals of industrial facilities, the maintenance-assembly-repair (MRA) procedures are indeed one of the most strategic research field for AR interfaces. These procedures require highly specialized human workers to cope with very complex tasks. One of the most adopted procedures relies on the use of paper-based instruction manuals, that is, the human technician has to continuously switch his/her attention from the industrial asset involved in the MRA procedures, lowering the attention and generating high cognitive loads.

To cope with these limitations, Interactive Electronic Technical Manuals (IETMs) can be employed to improve the efficiency of expert and inexperienced human workers [130]. However, also IETMs present some limitations: it has been demonstrated that IETMs are not entirely part of the technician-machine interaction process, only slightly reducing the task time and the cost of the procedure. Moreover, the IEMTs seem to be inefficient against high cognitive loads [175].

The AR interfaces can effectively overcome the limitations of paper-based instructions and IETMs [432]. MRA tasks can greatly benefit from the AR technology [174] by reducing the total cost up to 25% and improving the overall performance up to 30% [439]. The International Data Corporation foresees that the assembly procedures will attract investments in virtual technologies within the order of almost $400 million and it is expected that the investments in the maintenance operations will grow up to more than $5 billion by the end of 2021[2].

The virtual assets commonly employed in AR applications for MRA tasks provide aids, guidelines or suggestions to the technicians. These assets vary from 3D virtual contents, which describe the procedure with animations, to virtual textual labels that describe the industrial asset or the related instructions. The computer generated contents are usually overlaid or positioned very close to the real item to be maintained, allowing the human workers to clearly visualize the virtual instructions and the real item at the same time. Furthermore, more advanced AR applications support telepresence systems providing remote technicians with the ability to actively support the local human workers during the most tricky steps of the maintenance procedure. Feiner et al. [113] can be considered among of the first pioneers in the development of AR applications for simple MRA tasks. Then, during the '90s, the researchers started to rigorously analyze the benefits of AR in this particular context [308, 322]. We have to wait until the beginning of the XXI^{st} century for the first example of tele-maintenace AR system, which allowed a remote technician to provide instructions to a local user by means of augmented assets [167]. Due to their intrinsic capabilities of not forcing users to stand stationary in a single location, handheld AR solutions have been also deeply investigated and exploited, kicking off the so called *mobile* AR [166, 338, 380]. As an example, in [210] a handheld AR application has been developed to improve the management of the constructions and facilities life-cycle. Other examples can be found in [390], whose authors firstly investigated multimodal interactions in AR environments by mixing voice commands with virtual pointing devices, or in the project MOON [393] developed by the AIRBUS Military. Information and 3D data were generated from industrial mock-ups and used to create assembly AR instructions for the aerospace industry.

As explained before, the AR interfaces proved to be very useful in the MRA

[2] https://www.idc.com/getdoc.jsp?containerId=prUS47012020

context. However, they also present some limitations that prevent their spread and dissemination. First of all, although the number of mobile devices has increasingly grown during the last decades, improving the research regarding the handheld AR interfaces, the technicians usually cannot be forced to hold a device in their hands because they need their hands free to perform their tasks. Moreovoer, video see-through devices are negatively affected by an intrinsic delay in the data transmission that may generate hazardous situations. Secondly, there are still no clear and reliable guidelines to develop AR applications for the Industry domain, preventing the spread of AR in maintenance and repair tasks. There are however some researchers who are trying to tackle these limitations. As an example, Manuri et al. [279] analyzed the effectiveness of several distinct markerless tracking systems, providing useful insights for the industry domain.

2.1.2 Training

The use of AR technologies for training purposes is specifically connected to the MRA tasks, as they are typically the focus of the user's training in the field of industry. AR strategies have been extensively explored over the years to strengthen conventional learning approaches, as teachers, instructors and trainers are constantly looking for new methods to improve their students' learning experience and to establish creative learning and training routes. Multimedia contents should not just offer an improved sensory experience that can boost the user-machine and user-to-user experiences, they can also enhance the reader's or viewer's motivation and interest [474].

Different studies explored the reasons underlying the maintenance-related procedural error reports, showing that certain maintenance errors are not attributable to the lack of proper task knowledge, and AR is considered to be a powerful tool to help task execution because of its intrinsic ability to improve the user's appreciation [454]. Another major advantage of the use of the AR technology for training is that AR allows students to model risky or dangerous activities or even disruptive events without risk.

The first examples of AR tools for supporting and educating technicians via virtual instructions can be traced back to the early 1990s [308, 322]. AR technologies are being used to train and assist human workers in a wide variety of sectors, such as manufacturing plants [151, 334], aerospace [80, 81] and automotive industries [416, 470]. Sanna et al. [377] proposed a scalable AR-based training framework for industrial maintenance to overcome the challenges of producing AR assets for teachers and instructors. The method provided the teacher the ability to create an AR-based training process, easily "tuning" its difficulty according to the abilities of the students. In addition, by using an interactive telepresence system, the instructor can provide remote assistance: the teacher has a feedback on what the student's camera frames, he/she can communicate with the student and

he/she can determine which stage of the process prevents the user from completing his/her assignment. Eventually, the instructor may also change the method, sending a revised version back to the students.

2.1.3 Product Control Quality

A difficult task to achieve is the development of a new industrial product. The manufacture of goods goes through many stages, such as conception, design and actual realization. Once a product is created, the inspection phase checks both that no mistakes occurred during the development process and that there are no inconsistencies between the original project and its actual realization. The overall procedure should be applied as quickly and reliably as possible for efficiency purposes. Both in the management of the business and in the actual production of the product, there is an increasing inclination to reach a level of perfection. Quality controls are strong at the end of the supply chain in order to launch effective goods on the market that better suit the requirements of the end consumer.

As far as commercial properties are concerned, goods are visually inspected using a list containing unacceptable product defects. This practice is generally referred to as *inspection*. Inspection may be carried out in various fields, such as agriculture, industry, government, mechanics, and consists of an organized inspection of a specific equipment or procedure. As the number of items and their data grows, the task of inspection becomes more complicated. Because of the cognitive weaknesses of human inspectors, the inspection may thus become less successful. In setting up the inspection process, AR clearly appears to be a promising technology, since it allows for a direct comparison between the actual object and the idealized one. Indeed, the operator can directly see a 3D image of the ideal object superimposed on the product that is being inspected. This method is sometimes referred to as *discrepancy check* [473].

Ramakrishna et al. [357] suggested an AR method used to inspect an industrial product. A printer is examined using some Android devices (Cardboard with cell phone, Google Glass and Tablet) that can extract object information by recognizing a QR code placed close to the printer. The collected information reported some printer specifics (type, year of manufacture, history of inspection, etc.) along with a checklist to be carried out during the inspection. On the user device screen, directions and manuals are shown afterwards, so the inspector can complete the procedure with all the necessary details. In the strategy suggested by Chung [67], an AR tool is presented to examine some small industrial items. The aim is to understand which is the most effective way to inspect a real product. Four distinct inspection modalities are evaluated and compared. Results show that, by being the quickest solution, the AR method offers the best efficiency. Furthermore, because the operator has to perform fewer tasks than with the other three modalities, the AR method displays the least number of errors. Finally, an AR framework that can

create 3D versions of real objects in real time, allowing for an instant inspection, is proposed in [473]. The algorithm identifies a given object's geometry and compares the 3D model with the actual one. The anomalies are measured with a precision of 0.01m.

2.1.4 Building Monitoring

Checking the construction process is a complicated challenge, it is vital to project management to detect actual or potential schedule delays in field construction activities. The main current methods present some drawbacks. The building data are few and usually manually collected. Moreover, the process of building monitoring is usually represented with quite complex visual metaphors. The AR interfaces can overcome these limitations by visualizing the construction process directly on the actual environment and any deviation from the original plan can be detected. Golparvar et al. [146, 147] proposed an AR system that superimposes the virtual building model over time-lapsed photographs. The software evaluates whether there are discrepancies between what is being constructed and what has been planned. If any deviation is detected in certain regions, the related virtual assets are highlighted with red color, whereas the 3D model is colored in green if the construction is proceeding as planned.

Verification and control procedures are applied until the actual facility is constructed to check if the end product is different from what has been designed. There are several ways to check an environment: the standard method is to verify by hand, using geodetic devices and laser scanners. The key drawbacks are due to the lack of an automated mechanism that converts the measured points of the instruments (laser scanner, etc.) into a 3D model that can be contrasted with the actual environment in situ. These limitations can be solved thanks to the ability of AR to be used in the real world. AR is being used by many projects [141, 209, 248] to boost the recognition of pipe system issues. In [141], a framework that can recognize specific environmental features to identify any pipe configuration issue is proposed. The application superimposes the 3D CAD pipe model over the actual pipe and thus the final user is able to identify any inconsistencies between them. Finally, Zollmann et al. [507] presented an AR interface combined with an Unmanned Aerial Vehicle system. The aircrafts capture aerial images that are combined in real time with the virtual representation of the construction site.

2.1.5 Human-Robot Collaboration

The AR interfaces are a promising technology that can greatly improve the users' ability to understand several robotic features, such as the movements of the mobile robots and robotic arms, the forces applied by an end-effector, the robot intentions etc. Instead of using human workers, industries often use Automated

Guided Vehicles (AGV) for material transport. AGVs usually follow a predefined path that makes it easy for employees to predict the intentions of the robots. However, it imposes certain limitations on the type of task that the AGV can perform. Since it is expected that they will be able to autonomously compute the best path in the forthcoming facilities, there is a need for systems that can help human workers in understanding the robot intentions to avoid any possible hazard. The robot must express its forthcoming intended movements explicitly in order to make the system safe: since sight is one of the most advanced human senses, it is clearly a great decision to make explicit an intended movement through a visualization tool. This approach has been used by several works: for example, a standard projector has been added to an AGV to display the robot path on the floor, providing the technicians the ability of foresee its upcoming movements [55]. Similarly, the motion of a robotic arm can be easily detected and visualized using an AR interface. In [4], the AR interface highlights not only the object that will be picked by the robot but also the related trajectory. Although visualizing information about the purpose of the robot may foster the human-robot collaboration, it is also important to consider *when* the virtual assets should be displayed. Regarding this particular topic please refer to [373]. Finally, robot force is also considered in [283], where the force component of the end-effector can be visualized using a handeld AR device. Furthermore, according to the force intensity, the components are highlighted using different colors.

2.1.6 Conclusions

This section investigated the different uses of the AR technology in the Industry 4.0 domain. Five major areas have been analyzed and discussed. Considering the complexity of each domain, this dissertation will focus only on the use of the AR interfaces for the collaborative robotic. In the next sections, this particular topic will be deeply investigated, showing the strengths and weaknesses of the AR interfaces. Moreover, it will be also discussed what has been done to improve their usage in this specific domain.

2.2 AR Interfaces for Collaborative Robotics

In order to identify strengths and weaknesses of the AR interfaces for the collaborative robotics, it is necessary to clearly define the concepts of *industrial robot*, *collaborative robot* and *collaborative operation*. If the concept of "industrial robot" is very well known and it can be defined as *automatically controlled, reprogrammable, multipurpose manipulators, programmable in three or more axes, which can be either fixed in place or mobile for use in industrial automation applications* [196], the terms "collaborative robot" and "collaborative operation" are instead less well-known and

there are some misconceptions that should be clarified. The term "collaborative robot" usually refers to "a robot that can work side by side with humans". Since this definition is indeed too ambiguous to provide a clear detailed explanation of the human-robot collaboration concept, a more appropriate definition is given. The term appeared for the first time in the ISO 10218, part 1 and part 2 [193, 194], along with the terms collaborative operation and collaborative workspace. However, it only describes the basic guidelines for the usage of the industrial robots: "*[. . .]. It describes the basic hazards associated with robots and provides requirements to eliminate, or adequately reduce, the risks associated with these hazards*". The detailed definitions of the collaborative operation and robot can be found in the ISO/TS 15066 [195]. It defines the "*safety requirements for collaborative industrial robot systems and supplements the requirements and guidance on collaborative industrial robot operation given in ISO 10218 1 and ISO 10218 2*". The defintions of the aforemtioned termas are the following:

- "*A collaborative robot is a robot that can be used in a collaborative operation*";
- "*A collaborative operation is a state in which purposely designed robots work in direct cooperation with a human within a defined workspace*";
- "*A collaborative workspace is a workspace within the safeguarded space where the robot and human can perform tasks simultaneously during production operation.*

Furthermore, it is possible to define a set of collaborative operations that can be carried out in the human-robot workspace, the so called *collaborative workspace* (CWS). An operation, to be considered as "collaborative", has to follow one or more of the following guidelines [195] (Fig. 2.1):

1. "*Safety-rated monitored stop*": if the worker is in the CWS, the robot cannot move;
2. "*Hand guiding*": the human operator controls the robot by an input device;
3. "*Speed and separation monitoring*": as the distance between the robot and the worker reduces, the speed of the robot reduces too;
4. "*Power and force limiting*": contact between the human and the robot is allowed.

One of the most important consequences is that the specific task and the working space determine the collaborative operation, not the manipulator itself.

Given the definitions of collaborative robot, operation and workspace, several papers have been collected and categorized (for the complete set of paper, see Appendix A), identifying three different macro-areas concerning the use of the AR

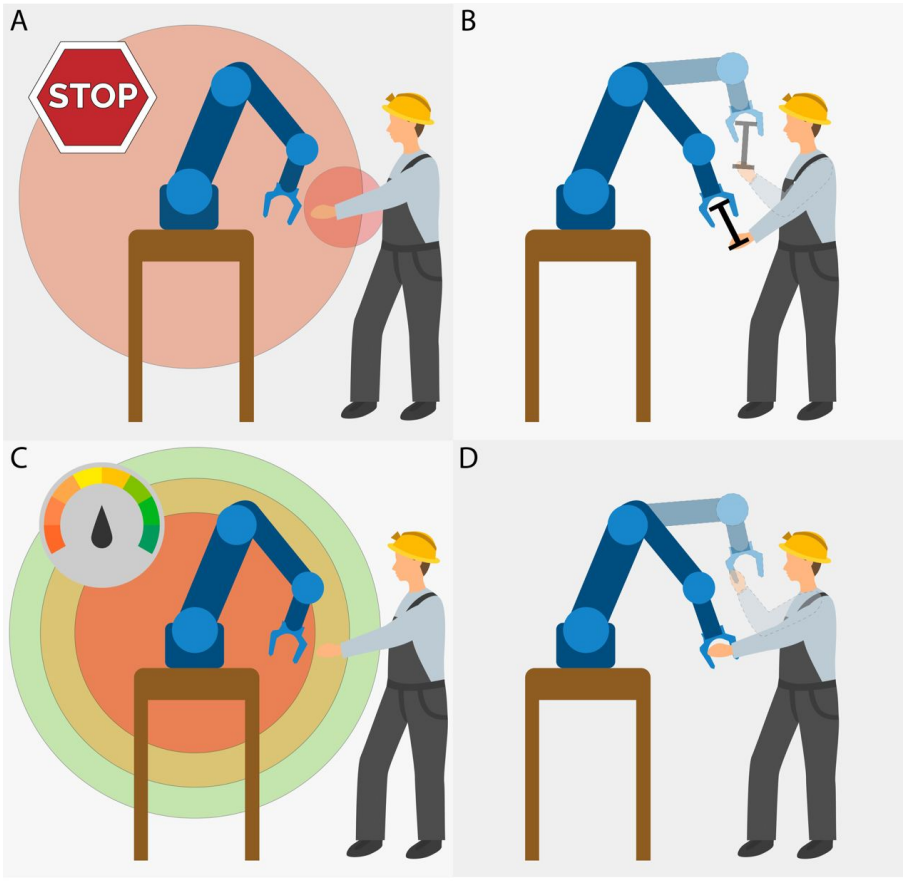


Figure 2.1: (A) the safety-rated monitored stop, (B) the hand guiding, (C) the speed and separation monitoring and (D) the power and force limiting guidelines.

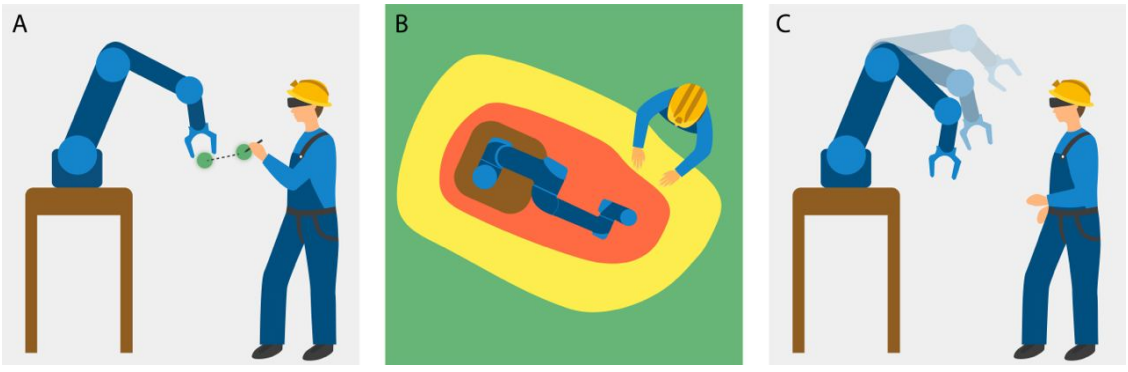


Figure 2.2: The Control Feedback, Workspace and Informative categories.

interfaces in the HRC domain: the Workspace, Control Feedback and Informative macro-areas (Fig. 2.2).

In the following sections, each category will be discussed, highlighting the main advantages and drawbacks of the modern AR systems and discussing what has been done to overcome such limitations.

2.2.1 Workspace

Works that employ AR assets to visualize the operative zone occupied by the robotic arm are listed in the Workspace group. The primary goal is to ensure a secure working environment, highlighting potential areas of collision with the robot. This macro-area can be further divided into two sub-categories:

- AR interfaces for large-size workspace and robots.
- AR interfaces for small-size workspace and robots;

These two categories will be discussed in the following sections.

Large Size Environments and Manipulators

There are some works that explored the use of the AR interfaces to collaborate with high-payload industrial manipulators positioned in fenceless worspaces [274, 294, 335]. In [294], a human operator can work together with a robotic arm by using a manual guidance system and a smart-watch interface. Furthermore, by using a wearable AR interface, the 3D robot workspace can be visualized using different colors (red and green) that highlight the robot working area and the safe working area of the user, respectively [274]. A collaborative automotive assembly scenario has been chosen as a use case and the main outcomes show that the proposed AR interface allows to considerably reduce the task time, passing from a 92.15 seconds to 76.31 seconds. A further development of this work can be found in [335].

Other approaches consider instead the use of 2D projected systems that do not force users to wear or hold in their hands any specific device. Two methods are proposed in [459, 458]. A tactile floor and a projection system have been combined in [459] to create a safe collaborative workspace. The tactile floor continuously checks the position of the human workers and transmits the related data to the projection system. The projector displays three different static zones using green (free zone), yellow (warn zone) and red (critical zone) colors. The system has been improved in [458] considering in real time the movement of the robot and changing the shape of the safety zones accordingly. Moreover, the speed of the robot movement varies according to the manipulator-human distance until the operator enters the critical zone and the manipulator movements are immediately stopped.

Small Size Environments and Manipulators

Vogel et al. [463] presented a dynamic safe AR interface. A camera is used to detect the reflected light beam, projected on a planar surface. The projected light defines a dynamic 2D workspace that can vary its shape over the time. The suggested AR interface was adopted to track and display the area occupied by a medium-sized manipulator [461, 460, 462]. By exploiting the principle of the light barriers, the projection system proposed in [461] is capable of detecting violations of the safety area in less than 125ms. In a further development, the robot 3D bounding-box is computed and projected on the workspace, thus allowing human operators to easily detect the hazardous zones [460]. Finally, the area surrounding the object to be manipulated during the collaborative task is highlighted in [462], providing a safer working zone (Fig. 2.3).

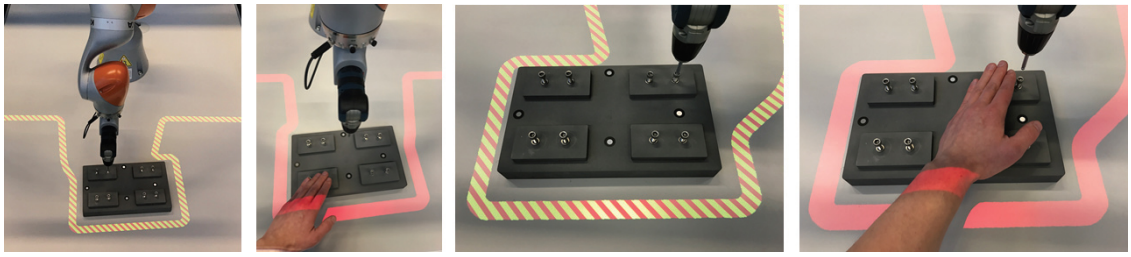


Figure 2.3: If the human worker enters the operative area while the robot is in motion, the manipulator suddenly stops moving and the projected safety zone is highlighted in red. Figure published in [462], licensed under CC BY-NC-ND 4.0.

2.2.2 Control Feedback

This category regards the use of AR contents as a feedback on an active input. An *active input* is defined as: "*the action exerted by the user with the purpose of interacting with the industrial manipulator placed in the same workspace*". The virtual contents can be employed to have a feedback on:

- path: a list of connected poses created by the user;
- input recognition: a generic user input.

In the following sections, these two types of feedback are introduced and discussed.

Path

Different works used a fixed camera and custom probes to create virtual paths [324, 65, 109, 321, 106, 108, 105, 107, 330]. One of the first study related to the control

of a robotic manipulator using AR desktop interfaces and image-based probes can be found in [324]. To ease the creation of the path virtual poses, a heuristic beam search algorithm is presented in [65], along with the visualization of the virtual manipulator. The accuracy of the tracking approaches used in the above works has been verified in [109] and the results show that is bounded between 10 to 15mm. To further improve the generation of the virtual poses, a Piecewise Linear Parameterization algorithm is presented in [321], allowing to create smooth curves, not negatively influenced by the speed with which the users control the probe. Instead of using flat image probes, Fang et al. [106] proposed the adoption of a cube composed by six different images that improve the tracking during large rotational movements. The accuracy of a similar system has been evaluated in [108, 105], showing that by using a fixed camera positioned at 1.5m from the workspace it is possible to obtain an accuracy of 11mm. The system has been user evaluated in [107] by comparing limited (i.e., the emulation of a teach-in method) or full (full set of AR tools) AR modalities. The main outcomes prove that the full set of AR tools allowed the users to complete the considered tasks in half of the time with respect to the limited modality. Furthermore, even inexperienced users could easily interact with the manipulator using the full AR approach. An improvement of the tracking accuracy can be found in [330], resulting in an position error less than $\sim$ 4mm.

Other examples of AR interfaces can be found in [497, 365, 453]. Specifically, Zaeh et al. [497] showed that by using a custom probe and a projected AR interface it is possible to program a robot arm in less than one fifth of the time required by a classic teach-in method. The projected virtual poses can be further manipulated to digitalize the workpiece surfaces, generating virtual bounding boxes used for collision checking [365]. A similar apporach is used in [453] to control a manipulator during a grinding process of ceramic objects (Fig 2.4). For additional works regarding the AR projected interfaces refer to [435, 11].

Wearable AR interfaces can be also used to create AR virtual paths. One of the early approaches can be found in [158], where the Microsoft HoloLens is employed to control a robotic manipulator. Kyjaneka et al. [238] proposed a wearable interface to visualize the torques of each robot axis, whereas a combination of a custom handheld pointer and an HMD are used in [323] to define the virtual path. The authors suggest that by using the proposed approach, the time required to complete a welding task decreases, passing from 347s to 63s. Finally, Quintero et al. [352] combined a speech interface with an AR wearable one and the results show that the users could program a robotic arm faster and with less physical workload than a traditional kinesthetic approach.

(a) (b)

Figure 2.4: (a): a human worker is creating a projected AR path. (b): the related AR path generation. Images published in [453], licensed under CC BY 4.0

Input Recognition

An AR feedback can be classified as *implicit* or *explicit*. The former is used when the interface displays only a virtual representation of the real manipulator without a visual asset highlighting the input given from the user. On the contrary, the latter is used when the AR assets highlight the user input itself. Several works employ implicit AR feedback [233, 277, 407, 15]. In [233], an AR gesture based system is compared with an AR gaze-based one in a pick-and-place scenario. The outcomes show that the gaze-based interface required lower work load and less time than the gesture one. Works presented in [15, 407] discussed two similar AR interfaces to control the virtual representation of the real manipulator. The main difference resides in the control paradigm: the first one employs the Wiimote device whereas the second one a wearable headset.

The objects of interest can be highlighted by explicit AR feedback. Works in [124, 123] presented an AR handheld interface to control a robotic arm during a pick-and-place scenario. When the users select a real object by pressing on the tablet surface, a virtual representation of the object itself is overlaid on the real one. In the extension of the previous work, the authors compared the handheld AR interface with two other exocentric and egocentric interfaces [123]. Despite the outcomes do not present significant differences in the success rate scores, the egocentric and exocentric user interfaces presented meaningful differences in terms of the completion task time. In Fig. 2.5 a human operator is interacting with the AR assets using the proposed AR interface.

Other examples of AR feedback on an active input can be found in [239, 240, 188, 97, 337, 136].

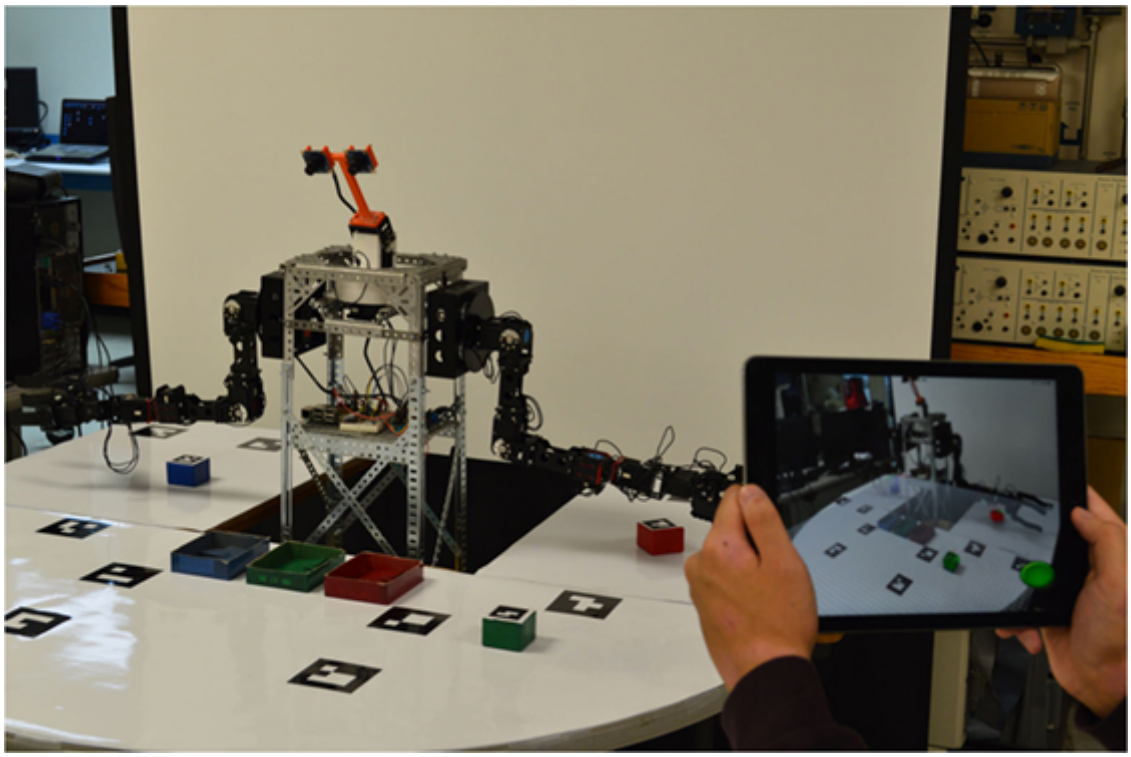


Figure 2.5: The virtual assets highlight the objects of interest. Figure published in [123], licensed under CC BY 4.0.

2.2.3 Informative

The Informative category encompasses works concerning the use of AR contents to highlight task or robot data. They have been divided in *task information* works, that focalize on employing AR assets to visualize generic task information, and *robot information* projects that employ the virtual metaphors to display data related to the robot itself (e.g., joint values or robot intentions).

Task Information

AR instructions can be categorized as *static/dynamic* and *interactive/not-interactive*. "Static" means that the spatial position of the AR content is context-independent, that is, its position cannot be changed. On the contrary, the "dynamic" term implies that their spatial positions can be changed depending on the context or on the user input. Both types can also be *interactive* or *not-interactive*, depending on whether the AR information can be changed by the users.

Examples of static not-interactive AR interfaces can be found in [274, 78, 294].

A wearable AR device is used in [274, 294] to display instructions and warning information. The information is fixed, positioned in the top area of the user interface in order to not interfere with the narrow field-of-view (FoV) of the wearable device. Different colors have been used (Fig. 2.6), depending on the type of information (green color for the assembly instructions and red color for the warning data). A similar user interface is proposed in [78]. Static interactive AR contents are instead employed in [97, 14] to improve the management of the task data.

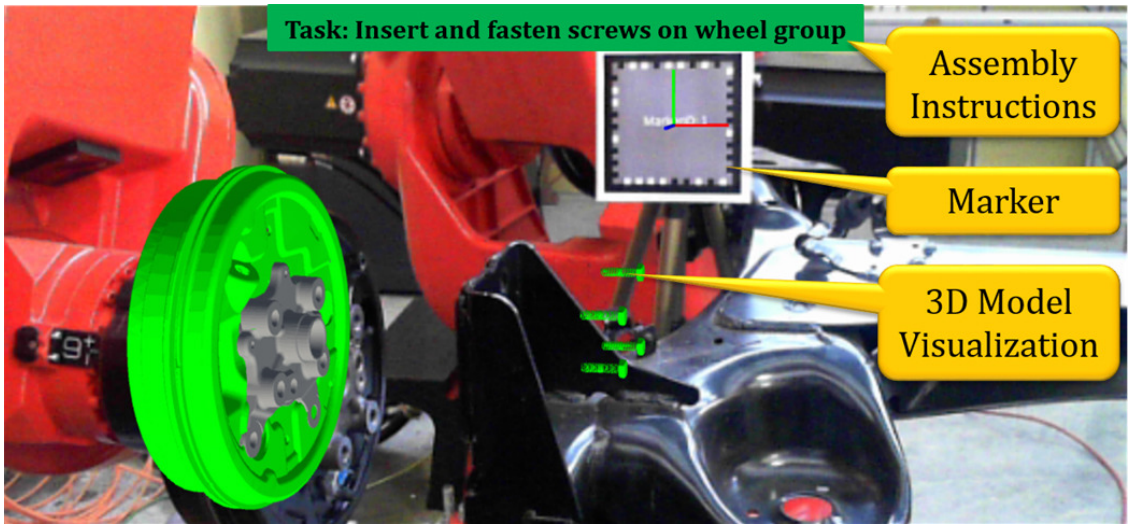


Figure 2.6: the top GUI area displays task information olored in green. Figure published in [274], license courtesy provided by Elsevier N. 5020740974415, Mar. 02, 2021.

Dynamic interactive interfaces are probably the most employed in the HRC context. Charoenseang et al. [58] used 3D virtual shapes to help users during an assembly task. A projected AR interface is presented in [249] to control the robot and the assembly process. The virtual assets can change their position depending on the step of the assembly procedure. Materna et al. [285, 284] proposed a combination of a projected AR interface with a touch enabled table. The projector displays the virtual contents on the table surface and the users can interact with them by exploiting the touching capabilities of the table. Other interesting works can be found in [437, 90, 68].

Robot Information

Robot components can be highlighted using AR assets. In [276] a comparison among the effectiveness of highlighting robot parts using different virtual metaphors (3D arrows, virtual leading lines and virtual text) is presented. The results show

that text and 3D arrows have been considered more effective than the leading line to highlight some specific areas of the manipulator. Similarly, Manring et al. [277] augmented the joint torques using different colors according to the intensity of the torque itself.

AR contents result to be a useful tool to display the future intentions of the industrial robots. In [12], a projector is employed to highlight the specific areas of the object that are going to be manipulated by the robot arm. The proposed interface has been compared with a desktop and a paper-based interface. The outcomes show that the projected interface was deemed more effective than the other two systems. However, the paper-based method has been considered more suitable to have an overview of the overall task. Wakita et al. [468] proposed an AR system that allows users to understand *when* the manipulator is giving attention towards their actions and *whether* their actions have been accurately understood by the machine. In [478], a methodology to determine the features required to optimally reference target objects is proposed. The robot expresses its intentions by means of virtual metaphors whose positions and orientations have been previously mathematically verified. Finally, Palmarini et al. [332] proposed a handheld AR system to display the upcoming motion of a robotic manipulator in an assembly scenario (Fig. 2.7). The main outcomes suggest that the users should be provided with context-awareness information to improve the safety perception. Interested readers should refer to [96, 56, 260] for additional works.

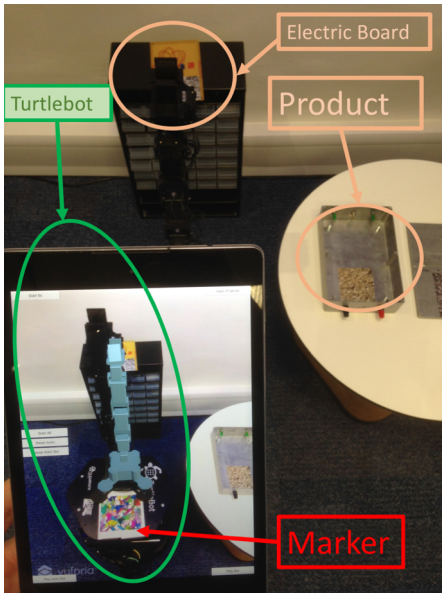
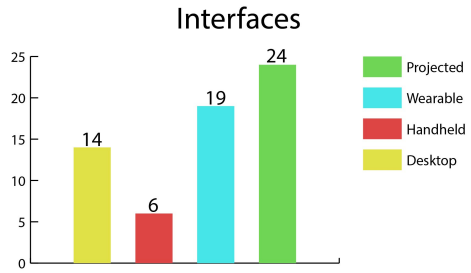
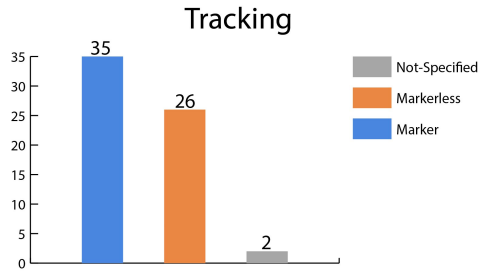


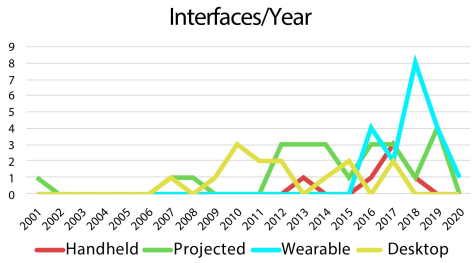
Figure 2.7: The augmetned robot representation can greatly help human operators in understanding the robot intentions. Figure published in [332], licensed under CC BY-NC-ND 3.0.



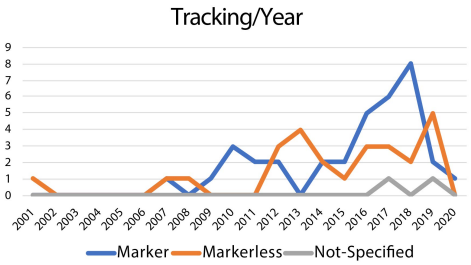
(a) The number of interfaces.



(b) The number of tracking approaches.



(c) The interfaces distribution over time.



(d) The tracking methodology distribution over time.

Figure 2.8: (a-c) The number of collected interfaces and their distribution over the time. (b-d) The tracking approaches and their distribution over the time.

2.2.4 Results

From the analysis of the presented macro-areas, it is possible to derive some considerations related to the use of the AR technology in the HRC context. The discussion will be presented answering three different research questions.

What are the main uses of AR technologies in the HRC context?

Figure 2.8 shows the spread with respect to time of the AR interfaces, along with the interface and tracking repartition. The projects AR interfaces are the most employed in the HRC context, followed by the wearable, desktop and handheld ones (Fig. 2.8a). These findings are not totally surprising: given their inherent ability to prevent users from wearing any specific device, it is fair to conclude that the projected interfaces have captivated researchers' interest, and therefore have been thoroughly examined and evaluated. Wearable interfaces, despite being the last to come to the market and only recently being used for testing, are increasingly becoming a hot research subject, not only in the HRC sector. On the contrary, the desktop interfaces are the "oldest" visualization interface, as well as one of the most well-tested and widely used. Nonetheless, since they require users to constantly turn their focus from the augmented environment to the real world,

they reduce the AR system's effectiveness. As a result, they could have been seen as less attractive in the HRC domain. Finally, handheld interfaces have some inherent limitations (such as requiring users to keep their hands occupied) that may have hampered their usage in the HRC scenario. These outcomes are supported by the distribution of the interfaces over the time (Fig. 2.8c). Although the paper distribution covers the 2001-2020 period of time, the analysis mainly focuses on the 2001-2019 period as very few papers have been published in the 2020 year thus biasing the collected results. It seems the desktop interfaces are increasingly less used whereas the projected and wearable ones are currently attracting the attention of the researchers. Considering the tracking methodologies (Fig. 2.8b), the marker-based techniques are slightly more employed than the markerless-based ones. These results seem be supported by the spread of the tracking approaches over the time, shown in Fig. 2.8d (also in this case, the analysis focuses on the 2001-2019 period of time). In fact, although the marker-based methodologies have been more used than the markerless-based ones, it seems that in the 2018 the marker tracking has abruptly stopped being employed in the HRC context. Markerless technology has typically been used by AR projected systems, and, since they are increasingly gaining the interest of researchers, markerless tracking is gradually becoming more embraced and employed.

Overall, the main considerations can be summarized as follows:

- the AR interfaces are mostly employed to (i) program and control a manipulator, (ii) display general tasks/robot data and to (iii) highlight the collaborative workspace;
- the projected systems appear to be the most encouraging ones;
- since the wearable interfaces have only recently appeared on the market, there is still room to foster the research in this particular topic;
- desktop and handheld interfaces do not appear to be acceptable for the HRC context;
- marker-based methodologies are still the most used but markerless approaches are increasingly capturing the researchers' attention.

What are the main strengths and weaknesses of the AR technologies in the HRC context?

The strengths and weaknessess can be derived by firstly analyzing which features and parameters are usually evaluated in the considered papers. The objective data that are normally considered are the following:

- task time;

- number of user errors;
- precision of the tracking.

On the contrary, the most evaluated subjective parameters are the following:

- usability;
- likability;
- workload.

Subjective data are either collected with standard or custom questionnaires. The most used standard questionnaires are the following:

- the System Usability Scale [48];
- the AttrakDiff [172];
- the NASA TLX [169].

The AR key benefit is that it reduces the amount of time it takes to complete a task. After analyzing the subjective results, it was discovered that users felt more at ease and fulfilled when engaging with AR systems than when interacting with conventional methods (such as kinaesthetic teaching or joypad control). Results that seem to be validated by usability outcomes, which show that AR systems received higher scores than the conventional methods. Finally, while the AR systems appear to minimize physical workload, the mental workload seems to be dependent on the interaction system (e.g. the mental workload may increase using AR systems combined with speech interfaces [233]). When it comes to flaws, the most pressing challenges are the tracking accuracy and the occlusion issues. The required accuracy varies by category, which has a significant impact on the system's performance (e.g. an AR control system may require higher accuracy with respect to an AR informative one). The occlusions, on the other hand, may cause the tracking system to fail, causing the virtual assets to vanish from the scene. Some questions have also been raised about the wearable devices' limited field of view, which prevents proper visualization of the augmented world.

Overall, the main strengths of the AR interfaces are the following:

- AR systems reduce the time required to complete a task;
- the users appreciate more the AR systems than the traditional approaches;
- the physical workload can be reduced using the AR interfaces, whereas the mental one depends on the input modality.

On the contrary, the main drawbacks are the following:

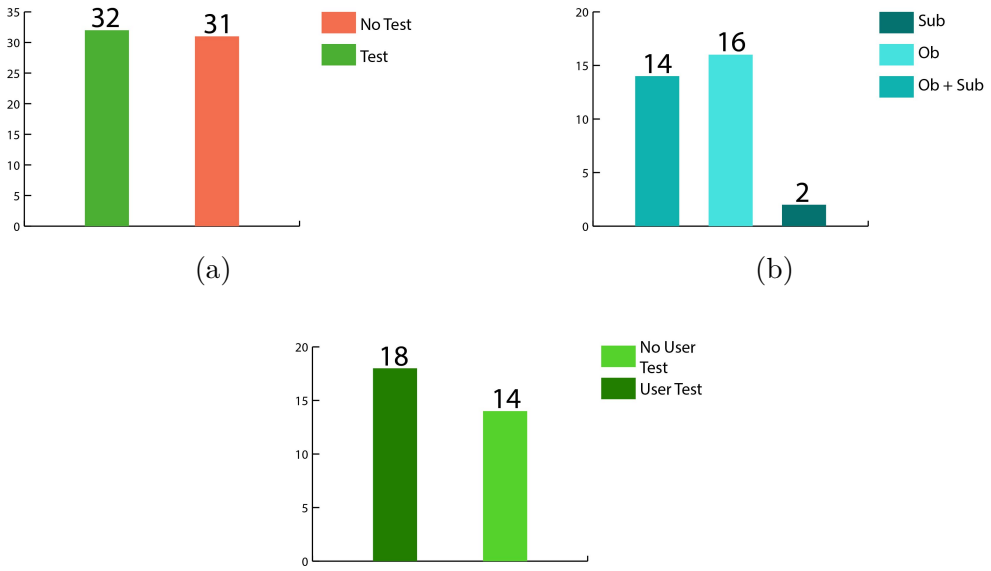


Figure 2.9: (a): only the 51% of the analyzed works have assessed the proposed AR interface. (b): the objective (OB) data have been more analyzed than the subjective (SUB) ones. (c): only slightly more than half of the papers with test results have also carried out user evaluations.

- accuracy of the tracking;
- occlusion problems;
- narrow FoVs of the wearable devices.

What are the potential future developments of AR technologies in the HRC context?

The first consideration is related to the evaluation process of the considered works. As shown in Fig. 2.9a, only 51% of the considered works have assessed the presented systems. Considering this 51% , 50% has assessed only objective data, 6% only subjective data and 44% both subjective and objective data (Fig. 2.9b). Moreover, only 56% of the papers has performed experiments considering inexperienced users (Fig. 2.9c). Only the work presented in [352] has evaluated the system considering also skilled users.

Nonetheless, it is possible to foresee the future developments considering the main outcomes used to answer the first two research questions:

- more user tests should be done to truly assess the AR interfaces (of any kind);

- wearable interfaces should be more researched in conjunction with industrial robots;
- the Workspace category has not been properly evaluated;
- the AR interfaces should be designed considering human-centered strategies.

2.2.5 Conclusions

This section presented the current state of the art related to the use of AR interfaces in the HRC context. Three distinct main macro-areas have been depicted, the Workspace, Control Feedback and Informative areas. Even though a fair number of papers have been analyzed, there is still room to foster the research in this particular topic, improving the chances that the AR technology will be soon employed on the manufacturing lines. Albeit some outcomes are in line with the actual state of the art, too few user tests have been done to verify the main strengths and drawbacks from a user perspective. It is fundamental that academics industrialists and researchers draw in a greater amount of users in assessing the effectiveness of the AR systems in the HRC domain. It is expected that the collaborative robots will strongly increase their presence in the Industry 4.0 context, replacing the "normal" robotic arm and integrating the efficiency of the machines with the adaptability of the human workers [340]. Human-centred strategies should be used to develop and assess the AR collaborative systems. The human operators should be taken as a reference during the design process to achieve a truly human–robot collaborative workspace.

Moving from these considerations, it will presented what has been done to improve the HRC context from a user-centred perspective. Specifically, it will be discussed how the detection of the robot faults can be improved using the AR interfaces (Sec. 2.3 and Sec. 2.4) and which virtual metaphors should be used during collaborative operations to foster robotic training procedures (Sec. 2.5).

2.3 A static AR Interface to display Industrial Robot Faults

As mentioned in the previous sections, there are several works that have investigated the effectiveness of the AR interfaces in the HRC context. However, but for the work in [96], situations in which the manipulators are affected by faults are not usually considered. In [96], an AR desktop interface is used to display the measurements of a set of robot sensors. The interface shows two distinct graphs giving an immediate feedback on the sensors' state. Despite the authors clearly detailed the advantages of the proposed system, the interface has not been evaluated by

alarm_count	fk_id_robot	alarm	severity	alarm_text
18	2213885	62006	11	Axis 1 Arm 1 motion under no servo control
18	2213885	60934	11	Axis 1 Arm 1 6052(3,4):short circuit in motor phase
13	2212894	60933	11	Axis 3 Arm 1 6048:axis braking, position error too high
10	2214565	58888	4	Arm 1 drive not ready for activation (sts:2)
6	2212894	62513	10	Axis 8 Arm 1 collision detected
6	2212894	62018	11	Axis 5 Arm 1 drive error (38006)

Table 2.1: This is an example of log fault file covering a period of two years. First column: the error frequency. Second column: robot id. Third column: the fault id. Fourth column: fault severity. Last column: the text-based description. Courtesy provided by the COMAU Italian company for the regional project HuManS.

user tests and thus it is not possible to verify its effectiveness. In the HRC context, human workers operate side-by-side with the robotic arms and unexpected faults may increase their anxiety because they cannot immediately understand which is the cause of the fault. Hence, an immediate video feedback could improve the operators' working conditions.

When the industrial manipulators are affected by faults, their movements and activities are immediately stopped for safety reasons, generating delay in the production process [60]. Nowadays, when a fault occurs, the following strategies is adopted:

1. the robot activities are stopped;
2. the description of the faults are saved in a log file (see Table 2.1);
3. the human operators, consulting the log file, try to solve the errors using technical manuals and their experience .

This procedure presents some limitations that should be overcome to ease the fault detection process. Firstly, the time required to solve the error is strongly related to the clarity of the text-based description and to the operator's experience. Secondly, not having an immediate feedback on the manipulator's internal

status, the users' trust in the industrial robots may decrease, compromising the effectiveness of the collaborative task. Hence, innovative detection strategies should be pursued to reduce the time required to detect and solve the unexpected errors and faults. The AR technology results to be effective to satisfy these requirements, directly displaying the virtual representation of the faults in the real environment (see Fig. 2.10). Hence, hereby a preliminary study carried out for this Ph.D. dissertation regarding the use of an innovative static AR interface is presented and discussed.

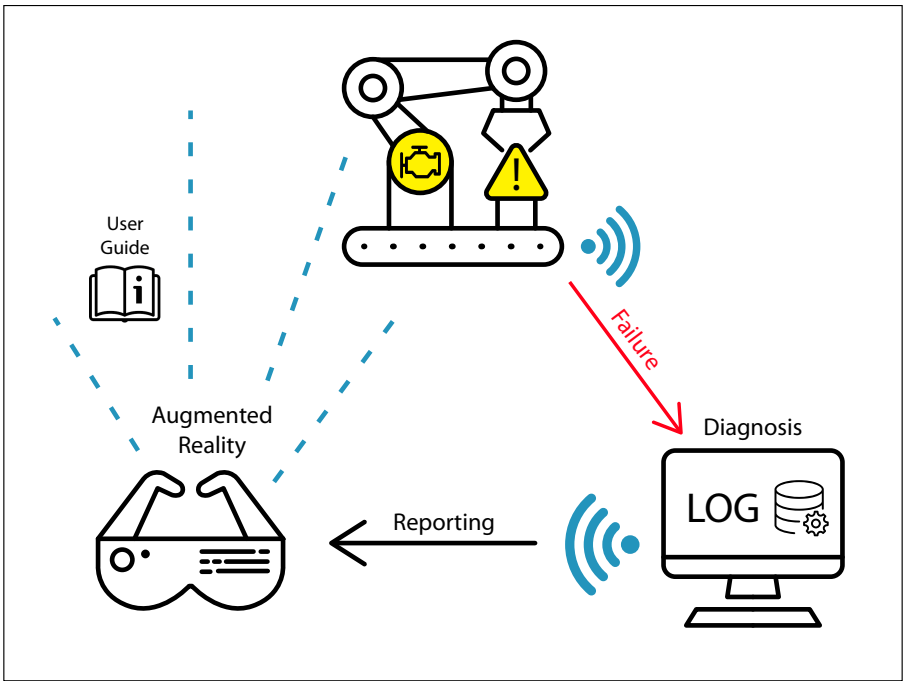


Figure 2.10: Thanks to AR, the robot fault can be directly visualized in the real environment.

2.3.1 Robot Fault Classification

Industrial robot faults can be classified in at least four different categories [110, 406]:

- sensor faults: although the physical quantity detected by the sensor is correct, the sensors can show to the users incorrect values;
- actuation system faults: motors and motor drivers faults;
- mechanical structure faults: collisions or brake faults prevent the correct functioning of the robot mechanical components (e.g., joints);

- overloading problems: the weight of the object to be manipulated exceeds the maximum payload weight.

To foster the sense of safety and reliability of the human operators, the faults should be detected and highlighted by means of virtual metaphors immediately recognizable by the human workers. Hence, the clarity of a set of virtual metaphors has been firstly evaluated by considering the following faults:

- velocity sensor fault: since velocity can be measured by a tachometer, a malfunctioning sensor produces a null velocity value;
- actuation system fault: the robot joints stop rotating around their axis;
- collision detection: the collaborative robots are equipped with sensors that can foresee unexpected collision, producing an abrupt stop of the robot movements. The human operators should be informed that the manipulator has stopped moving not because of an internal error but to avoid the collision;
- overloading fault: exceeding of the maximum payload weight.

2.3.2 Robot Fault Virtual Metaphors

Each of these faults has been represented by a 3D virtual asset with the following characteristics:

- a 3D circular arrow: it keeps rotating as long as the angular velocity sensor reads correct data. When the sensor reads null velocity values, it stops moving, changing its color. It is placed close to the robot joints;
- a 3D engine: if a fault on a joint engine is detected, this 3D assets starts blinking;
- a 3D sphere: the sphere surrounds the manipulator, highlighting its operative working-area. When a collision is foreseen, it starts blinking;
- a virtual anvil with a 3D warning signal: when the robot stops moving for overloading reasons, these virtual metaphors are overlaid close to the payload.

2.3.3 System Architecture

To evaluate the effectiveness of the virtual metaphors, a comparison between a wearable AR device (the Moverio-BT 200[3]) and a handheld one (a smartphone)

[3] https://www.epson.it/products/see-through-mobile-viewer/moverio-bt-200

has been done. Two different robots have been considered: a virtual representation of the Smart-5 Six COMAU manipulator and the real humanoid robot InMoov[4]. Although the real robot is not a industrial manipulator, its robotic arm is composed by different joints controlled by several electric motors. Thus, it has been considered suitable for testing the effectiveness on the virtual metaphors. The two robots have been used in four different scenes developed using the Unity3D[5] game engine with one main difference: at the beginning of each scene, the virtual robot is doing some pre-defined task and, after a certain amount of time, a fault occurs stopping its movement. On the contrary, the real robot is blocked as if an error has already occurred. Regarding the virtual manipulator, its movements have been generated using the Robot Operating System (ROS)[6] Kinetic version, running on a Ubuntu 16.04 personal computer (PC). The movement data are sent to the wearable and handheld devices using the rosbridge library[7]. Specifically, the PC acts as a server, creating a WebSocket server whereas the Unity3D applications act as WebSocket client. The server runs a simulation of the COMAU manipulator movements, thus generating movements identical to those of the real manipulator. When the clients connect to the server, the movement data are published on a specific topic, allowing to move the virtual representation of the robot in the Unity3D environment. The virtual robot and metaphors have been aligned in the real environemnt by using the Vuforia SDK and image target.

2.3.4 Test and Results

Ten users, with ages between 20 and 30 years, have been involved in the user tests. In order to evaluate the effectiveness of the proposed metaphors, the users had to visualize and understand their meaning in four different scenes (Fig. 2.11):

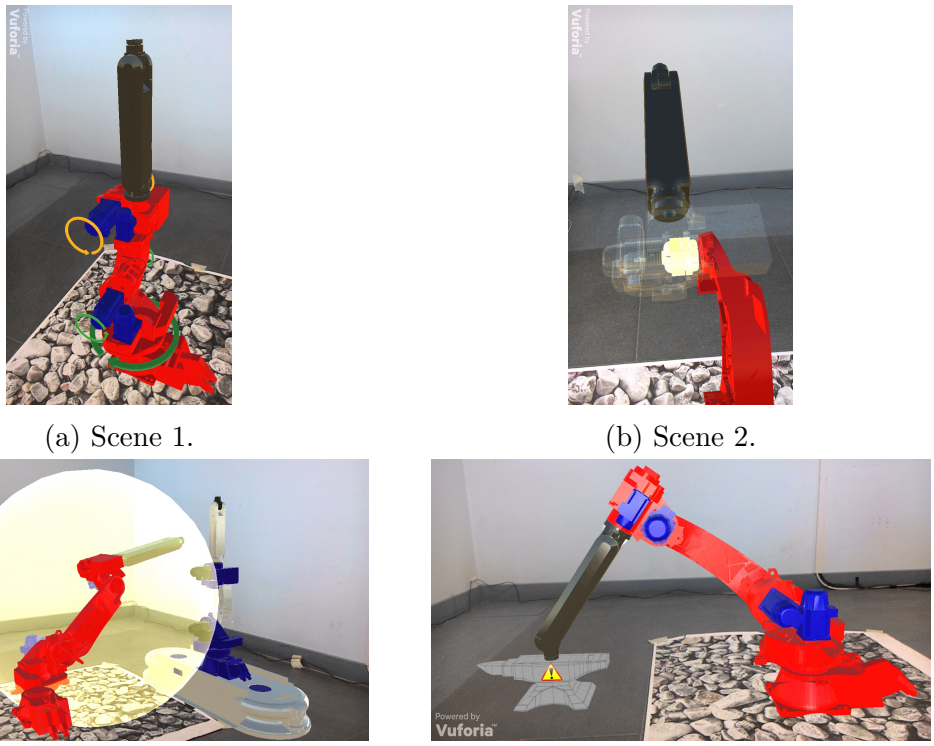
- Scene 1: velocity sensor fault. At the beginning the robot is doing a pre-defined movement and joint virtual arrows are rotating following the movement of the robot. Then, when the fault happens, the virtual manipulator does not stop its movements, because this type of error does not affect its motion but the the 3D arrows stop rotating, changing their color.
- Scene 2: actuation system fault. At the beginning the robot is doing a pre-defined movement. Then, when the fault happens, the virtual manipulator stops moving and the virtual engine of the blocked joint is highlighted.

[4] http://inmoov.fr/

[5] https://unity.com/

[6] https://www.ros.org/

[7] https://wiki.ros.org/rosbridge_library



(a) Scene 1. (b) Scene 2.

(c) Scene 3. (d) Scene 4.

Figure 2.11: The different test scenes.

- Scene 3: collision detection. An additional virtual AGV is approaching the robot arm and the virtual manipulator is doing a pre-defined task. When the manipulator foresees the collision with the AGV, it pauses its movements, letting the AGV pass, and the sphere starts blinking to highlight the risk of a collision. When the AGV has passed, the robotic arm resumes moving.

- Scene 4: overloading problem. At the beginning the robot is doing a pre-defined movement. After a certain amount of time, it tries to raise a payload that weights more than the robot's limit and its movement suddenly stops. A virtual anvil appears, superimposed on the payload.

For the experiments involving the real robot, the scenes were the same but there were only the virtual metaphors and the robot has been kept stuck for all the duration of the test.

For both the virtual and real robot, each user visualized two scenes with the AR glasses and the other two with the handheld device. The scenes were randomly selected. The users could inspect the whole scene trying to understand which was the nature of the fault. They could freely move in the augmented environment, watching the scene from different perspectives. After the test, the users had to fill

a questionnaire to verify their understanding of the virtual robot fault metaphors (the complete questionnaire can be found in Appendix C.1).

The detailed explanation of the main outcomes can be found in [85]. Hereby, a short summary is provided:

- scene 1: although 70% of the users deemed the 3D arrows suitable for representing speed, the velocity sensor fault has been confused with a joint block error;
- scene 2: 80% of the users understood the true fault nature;
- scene 3: 80% of the users understood the right cause of the collision;
- scene 4: 70% of the users understood the overloading problem;
- in the virtual robot experiment, the handheld interface has been considered more suitable than the wearable one;
- the narrow FoV of the wearable device did not allow to clearly visualize the virtual assets;
- in the real robot experiment, the users preferred the wearable AR interface.

The last results can be due to the number of the virtual assets of the augmented scene. In the virtual manipulator experiment, both the metaphors and the robot were virtual and the users faced serious problems in entirely visualizing the virtual assets. The narrow FoV of the wearable headset prevented from seeing them clearly, cutting most of the robot and of the virtual metaphors. On the other hand, in the other experiment, the robot was real and the users could focus on the virtual metaphors that are much less demanding in terms of FoV.

Moving from these last considerations, an adaptive AR interface to display industrial robot faults developed for this Ph.D. dissertation will be presented in the next section. The proposed system can display the virtual metaphors in positions always visible by the users, avoiding any occlusions and thus allowing to identify the robot fault in a much more intuitive way.

2.4 An adaptive AR Interface to display Industrial Robot Faults

The positioning of the virtual assets, or more in general, of the UI elements is a challenging and compelling problem. Only few works have analyzed this problem in the robotic context. Works presented in [90, 68] introduced two different methods for computing the best positioning of a projected AR interface employed

in a human-robot collaborative task. In [90], by continuously tracking the human operator pose, the system can project the AR interface in a position that is always visible from the user. On the other hand, Claassen et al. [68] introduced a method to find the most suitable planar surface for projecting an interactive AR interface used to control an AGV equipped with a robotic arm. The interface can be manipulated by the users using a background subtraction strategy proposed in [506]. The main outcomes suggest that if the projection angle exceeds 30°, the system fails to recognize the user's input. Although these projects present some interesting ideas, they are affected by some limitations that should be overcome to verify the true effectiveness of the proposed solutions. Specifically, no user tests have been done in [90], making it very hard to ascertain the system usability whereas, since the work in [68] did not compare the proposed solution with a non-adaptive interface, it is not clear which are the weaknesses and strengths of the projected interface. Moreover, no discussion has been done regarding the system behavior when no suitable planes are detected.

To tackle one of the main drawbacks of the wearable AR devices (i.e., the narrow FoV), this section will present an adaptive interface capable of displaying the virtual representation of the industrial robot faults in areas always visible by the user, using as input data the following parameters: (i) the pose of the reference object (i.e., the robot itself), (ii) the user's position and orientation and (iii) the FoV of the wearable AR device. To effectively manage the robot faults, the technicians should be able to clearly identify the manipulator's areas affected by the errors. The virtual metaphors should be placed in areas close to the fault's location avoiding to occlude the robot itself. Hence, the proposed interface adapts the assets positioning according to the user's movements, thus keeping the augmented assets always visible, properly oriented and correctly scaled. An image segmentation algorithm is introduced to place the virtual metaphors in areas not occluded by the robotic arm. By exploiting this approach, the human operators are able to immediately detect the faults, reducing the time and costs required to solve the problem. The adaptive interface has been compared with a non adaptive interface, similar to the one introduced in Sec. 2.3.

The overall methodology presented for this Ph.D. dissertation is composed of four different steps that will be introduced in the following sections.

2.4.1 Fault Representation

If the non adaptive modality (NAM) introduced in Sec. 2.3 employed arbitrary 3D virtual metaphors to represent the robot faults, the adaptive modality (AM) employed a more objective approach. Since usually errors and warning signals are represented by using 2D icons (e.g., smartphone battery warning, motor engine faults, etc.), it has been firstly determined a set of 2D icons that represents as much

as possible industrial robotic faults. To achieve the designated goal, a rigorous design approach has been adopted. Firstly, the 2D icon dataset has been generated determining the most relevant terms related to the robot faults. Using works presented in[110, 406], ten sentences related to the manipulator faults have been manually identified, determining ten different fault categories called *base_list_sentences* (Table 2.2). In addition, it has been verified that these categories could be found in the error log file provided by the COMAU Italian company (refer to Table 2.1 for an extract of the log file).

Sentence	**Type**
Fault on joint position sensor	Sensor
Fault on velocity sensor	Sensor
Fault on a current sensor	Sensor
Overload	Overload
Fault in a speed reducer	Mechanical Structure
Collision	Mechanical Structure
Fault in the brake	Mechanical Structure
Fault in the controller input/output board	Actuation System
Fault in a motor drive	Actuation System
Software error	Actuation System

Table 2.2: The list of the ten base sentences.

Afterwards, each item of the *base_list_sentences* set has been used as an input of the following procedure:

1. S1: prepositions and articles removal;
2. S2: synonyms generation;
3. S3: word permutations generation.

As an example, the fifth sentence of Table 2.2 will be used to explain the above procedure. The S1 step produced as output the sentence *fault speed reducer*. This new sentence has been then used in the S2 step to derive a set of new sentences (called *synonyms_list_sentences*), using the S1 output. Synonyms were obtained using the WordNet online tool[8] (Table 2.3 shows the *synonyms_list_sentences*).

Finally, last step (S3) has been applied. By considering each item of the *synonyms_list_sentences* as a set of n elements, it is possible to generate all the

[8] http://wordnetweb.princeton.edu/perl/webwn

fault	**speed**	**reducer**
fault	velocity	reducer
fault	speeding	reducer
fault	hurrying	reducer
defect	speed	reducer
defect	velocity	reducer
defect	speeding	reducer
defect	hurrying	reducer
flaw	speed	reducer
flaw	velocity	reducer
flaw	speeding	reducer
flaw	hurrying	reducer

Table 2.3: An example of *synonym_list_sentences*. Each column shows the synonyms of the first line words. In this case, the word *reducer* has no synonyms.

possible unsorted subsets formed by k items, with $1 \leq k \leq n$. The number T of unsorted subsets at a specific k is computed as:

$$T = \binom{n}{k} = \frac{n!}{k!(n-k)!}, \in k = 1, ..., n, \tag{2.1}$$

where n represents the number of words of a specific sentence and k the dimension of a specific subset. Then, it has been possible to query the online icon database TheNounProject[9] using each generated subset. In case an icon was found, it was saved in the corresponding fault category. The icons not strictly related to the ten fault categories have been manually discarded, producing a final dataset composed of 121 icons, not uniformly divided in the ten faults categories (see Table 2.4).

Since the icons were characterized by a great variety of styles and forms, they have been re-designed by applying the approaches introduced in [253, 251, 252]: (i) plane composition, (ii) negative polarity and (iii) border.

Ten different questions, corresponding to the ten fault categories, have been submitted to both COMAU operators (12 people) and university students (52 people). In each question, the users could indicate which icons best represented the manipulator faults by selecting one icon among those proposed. The option "none of these" has been added to indicate that none of these icons was suitable to represent the specific fault. Priority has been given to the technicians responses.

Figure 2.12 2D-column presents the final 2D icon dataset. Because neither COMAU operators nor university students have indicated a unique preference for

[9] https://thenounproject.com/

	Category	# 2D Icons
Q1	Fault on joint position sensor	19
Q2	Fault on velocity sensor	14
Q3	Fault on a current sensor	13
Q4	Overload	7
Q5	Fault in a speed reducer	15
Q6	Collision	8
Q7	Fault in the brake	8
Q8	Fault in the controller input/output board	14
Q9	Fault in a motor drive	10
Q10	Software error	13
	Tot	**121**

Table 2.4: The 2D-Icons column shows the number of collected icons.

the Q8 category, this category has been discarded. Finally, the 2D icons have been converted into 3D virtual animated assets (see Fig. 2.12 3D-column) by a graphic designer.

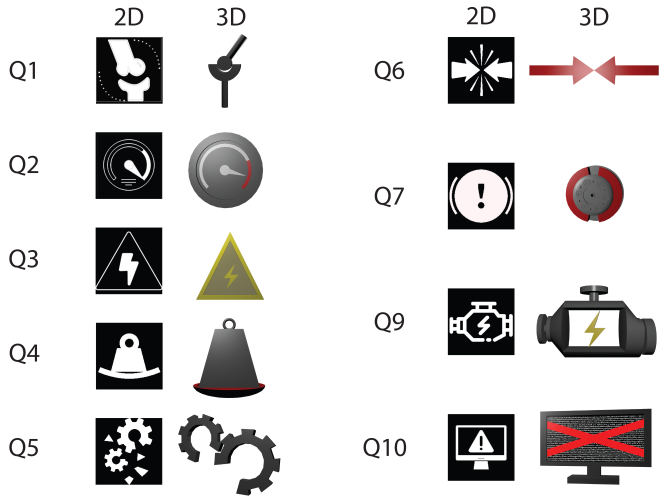
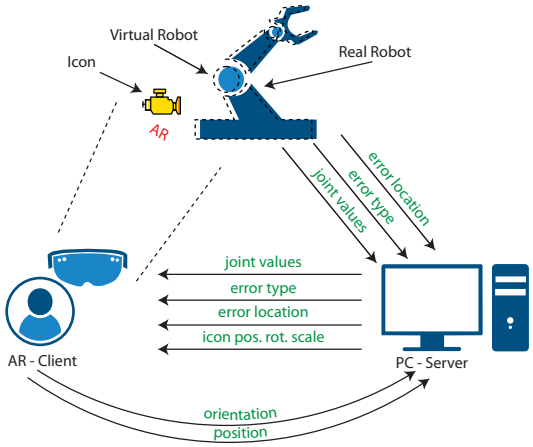


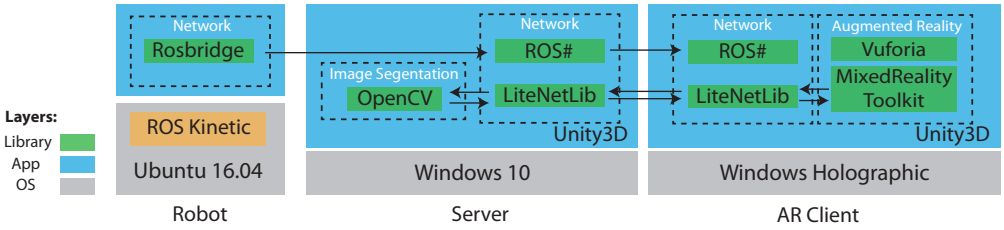
Figure 2.12: The 2D-3D conversion of the collected icons.

2.4.2 Fault Icon Placement

In order to explain the methodology underlying the icons placement, the hardware and software architecture are firstly introduced and detailed.



(a) The hardware architecture.



(b) The software architecture.

Figure 2.13: The system architecture.

Hardware and Software Architecture

The hardware architecture is composed by a manipulator, a remote server and the AR client. The Niryo One arm robot[10] has been used as a robotic manipulator. It is a ROS enabled robot platform and it provides a 6-DOF joint configuration and thus it can be used to represent an industrial manipulator. The AR client is represented by the Microsoft HoloLens HMD device. Using a wearable device, the technicians can keep their hands free to perform any possible tasks. Moreover, the icon visualization is independent of the environment (with a projected interface, the virtual assets may not be properly visualized if the projection surface is not planar). The server is a desktop PC that runs an algorithm capable of positioning the icons with AM or NAM modalities. The devices are connected on the same Local Area Network (LAN) through a User Data Protocol (UDP) socket connection. Figure 2.13 shows both the hardware and software architectures.

[10] https://niryo.com/

Implementation

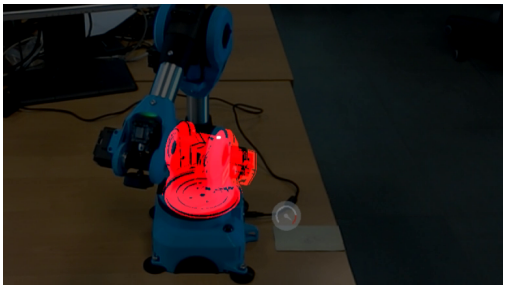
Regardless if acting in AM or NAM modality, the server receives from the real robot (i) an integer value between 0 to 5 indicating the joint affected by the fault (error location), (ii) an integer value between 0 to 8 indicating the fault type (error type) and (iii) the rotational values of the robot joints (joint configuration). The server uses these data to generate a faithful representation of the current state of the robot. In fact, the joint values are employed to animate a virtual representation of the robot whereas the error information is used for rendering the 3D virtual icons. The server forwards to the AR client the joint configuration and the error location and type. The joint configuration is again employed to animate another virtual representation of the manipulator. This virtual asset is kept invisible unless for highlighting the joint affected by the fault. Moreover, the server send to the AR client the position, scale and orientation of the virtual icon to be aligned with respect to the real robot. In the AM modality, the server further receives from the AR client the user's pose (position and orientation) every 20ms. These data are employed to control a virtual camera that stands for the user's actual pose. The camera's settings (FoV, far and clipping planes and resolution) have been set equal to the ones of the real AR client camera. Thanks to these data, the server can visualize its own representation of the virtual robot from the same point of view of the user.

The AR User Interface

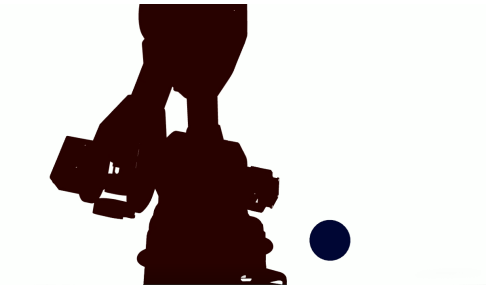
The AR user interface has been specifically designed to assist human operators who operate close to industrial robots. If a fault happens, the robot's activities are immediately stopped and thus the operators may not be aware of what is happening to the robot. Furthermore, when the fault occurs, users may not have their attention turned towards the robotic arm. To tackle these problems, a combination of computer generated contents and sound is employed to move their attention towards the manipulator. The virtual assets are represented by (i) the virtual Niryo robot, (ii) a virtual arrow and (iii) the virtual icons. When the AR client starts, the virtual manipulator is overlaid on the real one using an image target tracking system. The target is placed at a predefined distance from the real robot. The virtual robot mesh is kept invisible avoiding to occlude the real robotic arm. By exploiting the tracking information supplied by the Vuforia SDK (see Fig. 2.13b), the pose of the HoloLens can be derived determining the operator's position/orientation with respect to the manipulator. In case the technician is not looking at the robot when the fault occurs, a virtual arrow (Fig. 2.14a) moves the attention of the user toward the joint affected by the fault. When the user is close enough to clearly see the joint, the 3D arrow disappears. The joint affected by the fault is highlighted making visible the related asset (keeping invisible all the other joints) and its color is changed to red to emphasize the occurrence of a failure (see Fig. 2.14b). Furthermore, a

(a) The virtual arrow.

(b) The red meshes overlaid on the joint affected by the fault.



(c) The client view.



(d) The server view.

Figure 2.14: The augmented assets.

sound alarm is played to capture the technician's attention.

The Non-Adaptive and Adaptive Modalities

Regarding the icons positioning, the server can act in NAM or AM modalities. In the NAM modality, a set of predefined values are sent to the AR client. Regarding the position value, the icon has been positioned at a predefined distance k along the Z direction of the local reference system of the joint affected by the fault. k has been computed as:

$$k = 2L_{max}, \tag{2.2}$$

where L_{max} is the diameter of the manipulator's larger joint (see Fig. 2.15), equal to 12 cm. Therefore, any virtual icons can be positioned close to the manipulator avoiding to be placed inside the related joint. To determine the dimensions of the virtual icons, it has been empirically derived the minimum distance D_{robot} required to entirely visualize the manipulator using the HoloLens device, equal to 1.4 m. Then, some tests have been done visualizing the icons from that distance. In this way it has been possible to determine the dimensions (on the three axes x, y, z) of the icons, set equal to a value between 1 and k cm. Each axis dimension is strictly related to the shape of the icon (e.g., the joint position icon's height is larger than its width, etc.). Finally, once the size of all the icons had been established, a

constant and equal scaling factor S_c was assigned for all of them. Regarding the orientation, it is kept constant along the three axes, equal to $(0, 0, 0)°$.

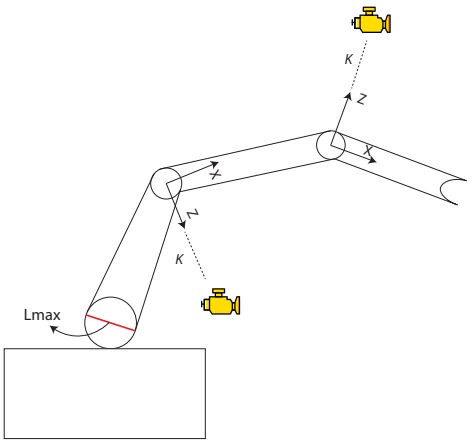


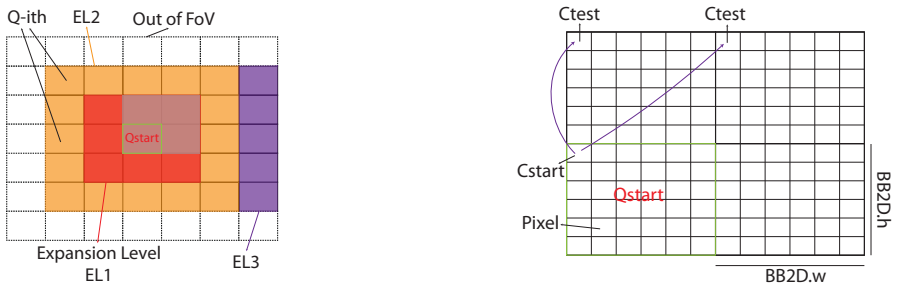
Figure 2.15: With the NAM modality, the virtual icon is placed at a fixed distance k along the Z axis of the joint local reference system.

In the AM modality, the server places the virtual icon in a position close to the joint, always visible to the user and not occluded by the real manipulator. By using the HoloLens position and orientation data, the server can visualize the robot from the user's point of view (Fig. 2.14c - 2.14d). This information is used in an algorithm, divided into three different steps:

1. A1: icon's scale factor determination;
2. A2: icon's position determination;
3. A3: icon's orientation determination;

During A1, at the fault time, the server instantiates the icon describing the fault in the exact position of the related joint, using the same pre-defined scale factor S_c used in the NAM modality. Since the correct position of the icon has not been computed yet, the virtual icon is kept invisible. The scale factor is updated at each frame using the following approach. Let $J(x_j, y_j, z_j)$ and $V(x_v, y_v, z_v)$ being the position of the joint affected by the fault and the position of the virtual camera in the world reference system. The distance D_{JV} is employed for computing the scale factor S_{icon} of the icon:

$$S_{icon} = (\frac{D_{JV}}{D_{robot}})S_c \tag{2.3}$$



(a) (b)

Figure 2.16: (a) The Q_s grid. The most external rows and columns are out of the FoV. EL1, EL2 and EL3 are the three Expansion Levels. (b) Four different Q_s. Each Q is represented with a pixel matrix defined by a Q_{start} and C_{start}.

If D_{JV} exceeds two pre-defined min and max values D_{min} and D_{max}, S_{icon} is no more updated to not excessively scale the icon. D_{min} and D_{max} can be empirically determined considering the collaborative workspace (in this case are equal to 35cm and 3.5m, respectively). Concerning A2, the following procedure is applied when the user changes his/her pose, that is, the virtual camera movement exceeds some predefined thresholds (in terms of position and orientation displacements):

1. B1: icon's projection on the 2D camera plane;
2. B2: thresholding and computing of the areas not occluded by the manipulator on the camera image;
3. B3: computing of the most suitable icon's position on the camera image;
4. B4: conversion from 2D to 3D coordinates.

During B1, the 3D icon's bounding-box (BB_{3D}) is projected onto the camera screen space using the world-screen transformation matrix, determining a BB_{2D} in pixel dimensions. BB_{2D} approximates the space occupied by the icon in the camera screen space.

Computed BB_{2D}, the camera image is analyzed and processed to determine a suitable area, large enough to contain the icon and not occluded by the manipulator (B2 step). The proposed approach uses a color thresholding technique to identify the areas occupied by the manipulator on the camera. In order to do so, the color of the manipulator's virtual model has been set to red and the virtual camera background to white. Hence, red pixels represent forbidden areas whereas white pixels identify possible suitable zones. Given a camera of MxN resolution (width and height, respectively), let I_{rgb} be an MxN matrix representing the RGB image acquired by the virtual camera. I_{rgb} is then converted in the hue, saturation, value

(HSV) color space, obtaining I_{hsv}. Applying an image segmentation algorithm on I_{hsv}, the thresholded matrix I_t is obtained as:

$$I_t(x,y) = \begin{cases} 255 & r_{min} \leqslant I_{hsv}(x,y) \leqslant r_{max} \\ 0 & otherwise \end{cases}, \tag{2.4}$$

where r_{min} and r_{max} are two constant values used to identify the red color and $I(x,y)$ identifies the value of a specific pixel in the corresponding matrix. Then, I_t has been subdivided in quadrants Qs of dimensions equal to BB_{2D}, generating a grid (see Fig. 2.16a).

Given a starting quadrant Q_{start}, the main goal of the B3 step is to find a suitable Q that minimizes the distance from Q_{start}.
Let EL_{max} be the maximum *Expansion Level* (Fig. 2.16a) determined as:

$$EL_{max} = max(Q_{r\_max}, Q_{c\_max}), \tag{2.5}$$

where Q_{r_max} and Q_{c_max} represent the maximum number of quadrants on the same row and column of Q_{start}, respectively, counting from Q_{start}. A Q-ith quadrant is defined by its upper-left coordinate C-ith. A quadrant to be tested Q_{test} is determined as:

$$C_{test} = (C_{start}.x + kBB_{2D}.w, C_{start}.y + uBB_{2D}.h), \tag{2.6}$$

where k and u represent two integer numbers used to access the Qs of the grid and C_{start} is the (x,y) coordinate of Q_{start} upper-left pixel (see Fig. 2.16b). In order to iterate over all the available Qs, the following equation has been employed to determine the k and u values of Eq. 2.6:

$$k = \pm e, u = -e + i, e - i, \tag{2.7}$$

where $e = 1, ..., EL_{max}$ and $i = e, ...,0$ are positive natural numbers. Depending on the ratio between $BB_{2D}.w$ and $BB_{2D}.h$, three different checking orderings exist (Fig. 2.17).

Established an ordering, the algorithm iterates over all the Qs of a specific s by first checking those that are at a shorter distance from Q_{start}. The assessment order of different Qs that are at the same distance from Q_{start} has been defined in advance. During the iteration, the algorithm controls the suitability of a quadrant to position the virtual icon. The evaluation process consists of evaluating the sum of all the $I_t(x,y)$ of a specific Q. If the sum is greater than zero, some red pixels have been found and thus a physical area of the robot is occluding that specific Q. Otherwise, a suitable quadrant $Q_{selected}$ has been found and promoted to be used for placing the virtual icon. Starting from $Q_{selected}$, the closest 3D position to the fault location should be determined (B4 step). Let V and J being two positions representing the camera and the joint affected by the fault in world coordinate

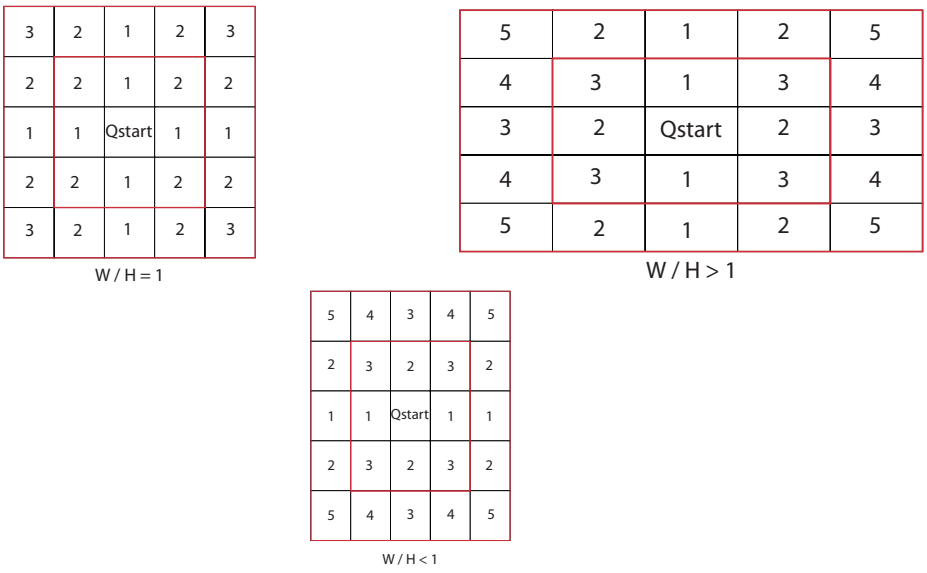


Figure 2.17: The three different checking orders.

reference system. From the central position of $Q_{selected}$, a raycast is projected and the ray point R is sampled, computing the vector $\vec{VR}$. $\vec{VJ}$ is projected on $\vec{VR}$, finding a 3D position not occluded and close to the manipulator (Fig. 2.18). Once a suitable 3D position is identified, the orientation of the icon is updated so that the forward vector of its local reference frame points towards the virtual camera position, continuously facing the user (A3 step). A detailed algorithm pseudocode can be found in Appendix B.

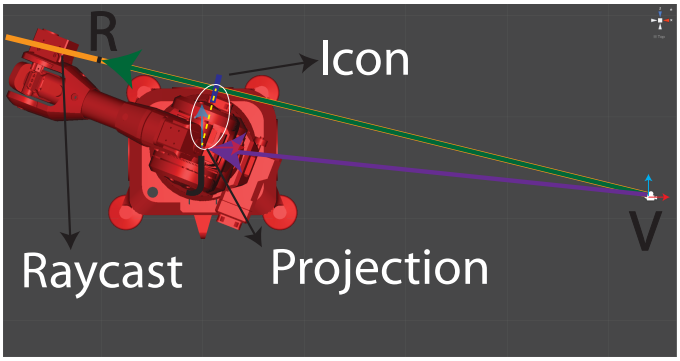


Figure 2.18: The 2D to 3D icon projection. V is the camera, the yellow line is the ray-cast and J the position of the joint.

The proposed algorithm can be adopted starting from any Q_{start}. The choice of Q_{start} is dynamically determined, taking into consideration the user's pose changing:

1. if the icon has not been placed yet (e.g., when the application boots) or it is

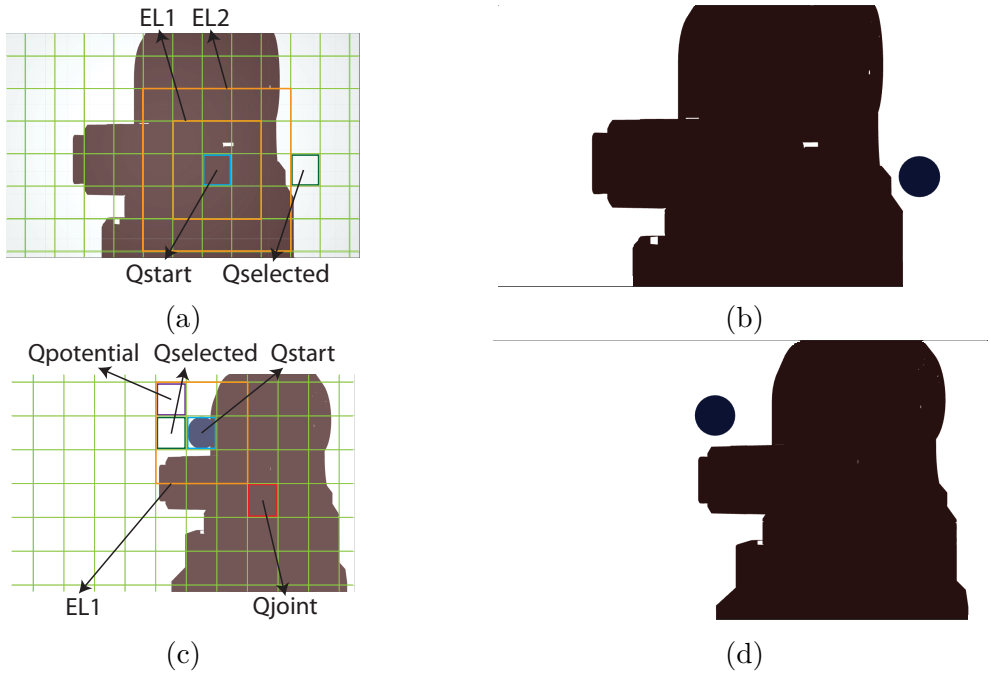


Figure 2.19: (a) The selected $Q_{selected}$ given a Q_{start}. (b) The icon positioning using the computed $Q_{selected}$. (c) During the computing of a new position, $Q_{potential}$ is discarded because it is further than the new $Q_{selected}$. (d) The icon positioning using the new computed $Q_{selected}$.

no more in the FoV (e.g., rapid changes of the user's pose), it is chosen the Q that encompasses the projection of the 3D position of the joint affected by the fault on the camera space (Fig. 2.19a - 2.19b);

2. if the icon is in the FoV, the Q that encompasses the projection of the 3D position of the icon on the camera space is selected.

In the second case, an additional verification is performed to verify whether the manipulator occludes Q_{start} or not. In case of an occlusion, the algorithm does not look for the first adequate Q of a specific Expansion Level EL, but it computes N adequate quadrants ($Q_{potential}$) of s ($N \geqslant 1$) and it is chosen as $Q_{selected}$ the one that minimizes the Euclidean distance from the quadrant that encompasses the projection of the 3D position of the joint affected by the fault (Q_{joint}). Hence, abrupt and undesired movements of the 3D icon are avoided and the icon is kept as close as possible to the fault location (see Fig. 2.19c - 2.19d). Finally, in case the joint affected by the fault is not in the FoV, the overall algorithm is not applied.

2.4.3 Experimental Tests

The comparison between the NAM and AM modalities has been carried out at Politecnico di Torino by involving 34 people, with ages between 19 to 30 years. The modalities order (AM-NAM, NAM-AM) has been changed every time to avoid learnability effects.

The tests have been planned simulating a fault condition during a human-robot collaborative task, considering also situations in which the technician is not paying attention towards the robot (i.e., the robotic arm is not in the FoV):

- nine users' starting positions (SPs) have been identified. The SPs have been determined taking into account both near and far positions from the robot;
- each tester starts the experiment from a specific SP, wearing the HoloLens device and giving his/her back to the robot;
- the real robotic arm is already stuck in the fault configuration;
- when the alarm sound informs the user of the occurrence of a new fault, the user can start freely moving around the environment, trying to identify which type of fault has occurred and on which joint in the shortest possible time;

Once the fault is recognized, each user starts from the next SP following the same procedure. Hence, each user has been involving nine different tests for each modality. To further avoid the learnability effects, at each SP the joints affected by the faults have been randomly picked out from the original number of robot joints (six for the Niryo robot). Only one fault could occur at a time and it has been randomly selected out of the original 3D icon set, discarding at each SP the corresponding virtual icon displayed before. Hence, all the possible 3D icons have been tested in each modality. In Fig. 2.20 the collaborative environment and the nine SPs are shown; the arrows stand for the user's starting orientations.

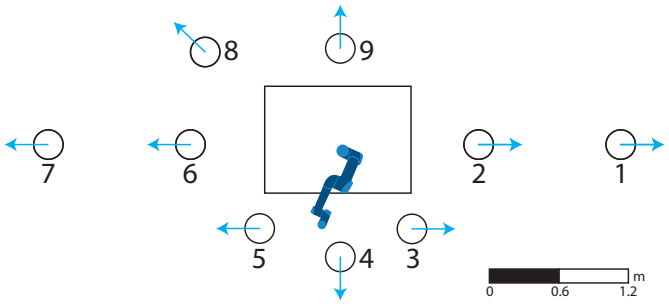


Figure 2.20: The nine SPs and the related starting orientations.

Before the experiments, each user has been introduced to the test, showing him/her the virtual icons (along with their meaning) and the Niryo manipulator.

Independently of the modality, both objective and subjective parameters have been evaluated. Specifically, the objective parameters are the following:

- the time required by the users to identify the fault and the related joint at each SP;
- how many times users have mistakenly recognized the faults and joints at each SP;
- the number of head translations and rotations carried out by the users at each SP.

The virtual icons could be identified by either describing their forms or by saying the corresponding names. The robot joints could be recognized by saying their numbers (from 0 to 5) or by pinpointing them. The users had to provide such information by voice to an external operator. Regarding the subjective parameters, a fifteen-sentences questionnaire divided in five different sections has been submitted to the users every time they completed the evaluation of a specific modality. The first section (QR1) regarded general information about the user, familiarity with AR and robotic arms and the users had to fill it before starting the test. Then, the remaining sections could be filled after having completed the evaluation of a specific modality:

1. QR2: clarity of the 3D icons. Whether the icons meaning was comprehensible by users;
2. QR3: perception of the 3D icons. Whether icons position, rotation and scale values were suitable from a user's point of view;
3. QR4: suitability of the employed FoV. Whether icons could be properly visualized with the device's FoV;
4. QR5: system's global score.

2.4.4 Result Analysis

In this section, the collected results are detailed and discussed.

The Objective Results

Objective results are related to the completion time, the errors and the number of movements required to identify a particular fault. Referring to the time and movement data, three distinct datasets have been collected, corresponding to the time, rotation and translation values. Since the virtual icons and the related joints have been randomly chosen at each SP, there is no correlation between a value

	Time (s)		Translations (m)		Rotations (°)	
USERS	**AM**	**NAM**	**AM**	**NAM**	**AM**	**NAM**
1	46	213	6	21.48	1827	3939
2	60	60	12.12	15	2095	2869
3	62	107	9.96	28.08	2094	4128
4	61	229	12.48	77.4	2261	7645
5	99	74	10.32	16.68	2158	2425
6	87	121	14.88	27.48	2953	3865
7	72	63	11.04	18.72	2109	2627
8	57	119	17.4	39	2074	3655
9	82	97	15.24	24.12	2628	3554
10	69	165	14.04	32.4	2400	4536
11	84	79	16.32	16.32	2422	2932
12	49	99	6.72	18.12	1755	3194
13	90	67	15.36	16.8	2207	2302
14	87	116	15.24	20.88	2429	3702
15	95	100	14.64	21.12	2620	3540
16	55	99	10.32	23.64	2236	3323
17	77	77	12.24	14.16	2353	2378
18	55	85	10.68	15.84	2026	2541
19	56	48	13.08	14.16	2098	2215
20	68	97	12.6	24.72	2022	3152
21	125	112	13.8	26.4	2285	4045
22	57	81	18	28.92	2580	3655
23	148	125	36.36	43.68	3810	3992
24	62	102	7.92	13.92	1651	2017
25	92	76	17.04	18.24	2607	2657
26	54	104	14.76	34.68	2157	3709
27	64	47	12.6	13.92	2140	1988
28	42	78	12.24	29.04	2129	3543
29	78	78	17.04	27.6	2624	3600
30	56	90	26.4	32.88	2571	3033
31	73	69	11.52	9.6	2062	2048
32	43	69	10.56	17.52	1890	2391
33	79	63	15.24	15.72	2248	2305
34	58	118	8.52	33.84	1928	3810
AVG	71.82	97.85	13.90	24.47	2277.91	3273.97
STD	22.65	39.86	5.49	12.38	392.28	1049.55

Table 2.5: The time, translations and rotations results.

USERS	QR2 CLARITY		QR3 PERCEPTION		QR4 FOV		QR5 SCORE	
	A	NA	A	NA	A	NA	A	NA
1	75	91.66	62.5	75	75	100	75	75
2	91.66	58.33	91.66	33.33	75	25	100	50
3	91.66	75	91.66	95.83	50	75	75	75
4	100	66.66	100	75	75	75	100	75
5	91.66	58.33	100	70.83	100	50	100	75
6	100	91.66	91.66	58.33	75	75	75	75
7	91.66	83.33	91.66	75	75	75	75	50
8	100	100	100	87.5	100	75	100	100
9	100	83.33	95.83	66.66	75	50	100	75
10	75	75	95.83	70.83	75	25	100	75
11	91.66	41.66	100	25	75	0	75	50
12	100	91.66	100	62.5	100	75	100	75
13	91.66	100	95.83	75	75	50	75	75
14	83.33	83.33	100	50	100	75	75	50
15	100	33.33	95.83	8.33	100	25	100	25
16	75	58.33	91.66	41.66	100	75	75	75
17	100	91.66	91.66	75	100	75	100	75
19	83.33	66.66	91.66	33.33	75	50	75	25
20	75	75	66.66	58.33	100	75	75	75
21	91.66	58.33	75	50	75	50	75	50
22	91.66	91.66	95.83	75	100	75	100	75
23	83.33	66.66	66.66	58.33	75	50	75	50
24	75	50	83.33	66.66	75	50	75	50
25	83.33	83.33	66.66	41.66	75	75	75	50
26	100	58.33	100	50	100	50	100	75
27	100	66.66	87.5	41.66	75	25	100	75
28	100	25	100	41.66	100	25	100	25
29	83.33	41.66	75	16.66	100	50	75	25
30	83.33	75	100	83.33	100	75	100	75
31	91.66	16.66	79.16	16.66	100	0	75	50
32	100	100	91.66	87.5	100	75	100	75
33	83.33	50	91.66	25	100	25	100	25
34	100	75	62.5	66.66	100	50	100	75
AVG	90.19	68.86	88.96	56.36	87.5	54.41	88.23	61.02
STD	9.05	21.54	12.13	22.40	14.10	24.20	12.66	19.64
Median	91.66	70.83	91.66	58.33	100	50	100	75
IQR	16.67	25	14.58	33.34	25	25	25	25

Table 2.6: The subjective outcomes normalized in the 0 - 100 interval. Refer to Appendix C.2 for the complete questionnaires.

obtained in $SP_i(AM)$ (the i-th SP used in the AM modality) and one obtained in $SP_i(NAM)$, with $i = 1, ...,9$. Hence, the values of each SP have been summed up for every user to compute a global score for each dataset. Then, the average score of a dataset has been calculated by dividing the sum of the total values by the number of users. In Table 2.5 the time, translation and rotation values are shown. The Wilcoxon Signed Rank Test has been performed to statistically analyze the collected data. The AM modality provided the users the ability to recognize the fault typology and the related joint in less time than the NAM modality ($p = 0.001$, effect size $d = 0.556$) and with a fewer number of head translations ($p = 0$, effect size $d = 0.839$) and rotations ($p = 0$, effect size $d = 0.848$), lowering the physical effort (the effect sizes are a measure of the "strength" of the differences among the average values [70])[11]. By placing the virtual icon in positions close to the joint affected by the fault, the users could recognize both the fault type and the joint number at the same time, without having to change frequently their position and point of view. Moreover, the automatic scaling and orienting mechanism has allowed to keep the icon in the narrow FoV of the HoloLens device, thus maintaining it clearly recognizable. On the contrary, the icons that could be only recognized from a specific side (e.g., both the velocity and break icons present a circular shape and they could not be recognized if seen from a lateral view) forced the users to change their point of view and position more frequently in the NAM modality. Finally, no errors have been detected during the recognition of the icons and the related joints. Hence, their design seems to be adequate for the proposed task. Furthermore, the correct recognition of the joints suggests that the adopted tracking modality has been deemed suitable to correctly align the augmented assets to the real manipulator.

The Subjective Results

Similarly to the objective data, QR1, QR2 and QR3 outcomes have been aggregated, summing the scores of the related questions for each user and calculating the average values (to improve the data readability, the values have been mapped in the 0 - 100 range). Since QR4 and QR5 were composed by only one statement each, the pre-process has not been necessary. Regarding QR1, the users reported to have occasionally used an AR application and to have little familiarity with the HMDs. The 26% has experienced with a robot arm and only the 6% knew the Niryo Robot. Table 2.6 shows the remaining subjective results. The Wilcoxon Signed Rank Test has been performed to statistically analyze the QR2, QR3, QR4 and QR5 outcomes. Referring to QR2 and QR3, the users had a better understanding of the icons' meaning ($p = 0$, effect size $d = 0.730$) and they have deemed more

[11]The effect sizes have been computed as $d = \frac{Z}{\sqrt{N}}$, refer to [448]

suitable the position, scale and orientation values ($p = 0$, effect size $d = 0.84$) with the AM modality. Since it could happen that the icons were cropped due to the narrow HoloLens FoV, the users may have found some problems in understanding their intrinsic meaning (fact that seems related also to the time spent to recognize the virtual icons). Result that seems to be confirmed by the QR4 outcomes ($p = 0$, effect size $d = 0.775$) which show how the versatility of the AM modality has allowed the users to lower the unpleasant effects of the HoloLens narrow FoV. Finally, although the NAM modality has been assessed as acceptable, the AM modality has been preferred for the proposed task in QR5 ($p = 0$, effect size $d = 0.819$).

2.4.5 Conclusions

By identifying the most common robot faults, a rigorous methodology has been used to figure out which 3D virtual metaphors best describe faults on industrial robots. An adaptive modality to visualize the virtual metaphors has been presented. The user's pose and wearable device parameters have been dynamically employed to place the virtual assets in positions close to the fault's location, always recognizable by the user, without occluding the manipulator itself.

To evaluate the effectiveness of the proposed solution, a comparison between the adaptive modality and a non adaptive one has been performed. The results show that with the adaptive modality the users could recognize the robot faults faster and with less movements than with the non adaptive solution. The ability of positioning the icons in areas always visible from the users has allowed to reduce the burdensome limitations of the narrow FoV of the HoloLens device.

2.5 Collaborative Virtual Training for Robotic Operations

Although in this dissertation the effectiveness of the AR interfaces in maintenance and training tasks has been deeply discussed (see Sec. 2.1), it is possible to find several works that have improved the traditional procedures by proposing remote collaborative systems. They are usually characterized by a remote skilled operator who provides instructions to a local unskilled technician by means of virtual assets. As an example, a remote expert operator can help a local user, indicating the real objects to be used during a maintenance procedure [43]. The local unskilled user records the real environment by using a wearable AR device and the related streaming is sent to the skilled operator who can add annotations and abstract virtual metaphors. A similar system is proposed in [299], which shows that these kind of collaborative systems can greatly reduce the time and costs required to complete maintenance procedures. Other examples can be found in [504, 221, 470].

In addition to the traditional virtual abstract metaphors (e.g., arrows, circles, etc.), the research is moving towards interfaces with the aim of improving the perception and the efficiency of the human collaboration adding the visualization of the human gestures to the augmented scene [152, 438, 411, 471, 494]. As an example, Yin et al. [494] shows that the visualization of human body parts during a maintenance task allows users to learn the procedure in a more natural way. Thanks to the technological improvements in the reconstruction of the human motion, it is possible to generate animated and realistic virtual human avatars and thus considerable efforts are committed to analyze the reactions of the human beings during an interaction with virtual agents. As highlighted by some recent works [201, 200, 230] and commercial applications[12], the adoption of virtual avatars is becoming increasingly researched and explored. To ensure that the avatar will be positively accepted by the local inexpert operator, the acting of the virtual agent and its collocation in the real world should be as realistic and credible as possible. Hence, for the remote operator it becomes essential analyzing the local operator environment from an independent point of view. In fact, it has been demonsrated that being point of view independent greatly reduces the task time improving also the user's confidence [430, 431]. The state of the art highlights two fundamental aspects: firstly, since the animations greatly improve the effectiveness of the abstract metaphors, a virtual avatar should be properly animated, allowing the remote user to accomplish several types of actions. Secondly, the remote user's interaction should be view independent.

The goal of the approach proposed for this Ph.D. dissertation is to analyze how a robotic training scenario could benefit from these aspects by comparing a traditional remote assistance, based on abstract metaphors, with an innovative one, based on a virtual avatar.

2.5.1 System Requirements

Starting from the analysis of the state of the art, an assisted training procedure is composed of:

- a local operator doing a task in a dedicated real workspace. The workspace is equipped with all the tool required to complete the training procedure;
- an instruction set for helping the user in completing the task;
- a communication channel between the local and remote user.

Based on the above requirements, the following systems has been developed. A local unskilled operator (trainee) has to be trained to perform a task by a remote skilled

[12]https://objecttheory.com/prism

operator (trainer). Both the trainer and the trainee are not in the same physical environment. The trainee's real workspace is equipped with the tools needed to perform the task and the trainee can visualize the virtual instructions provided by the trainer (the common space composed by either virtual and real objects and accessible from both operators will be referred as *shared environment* SE) using a wearable AR device. The trainer can access SE through an immersive VR interface. Finally, the two operators are able to communicate through a bidirectional audio channel.

Since in the local AR environment all the virtual assets are aligned with respect to a known target, it is reasonable to employ such target as a shared reference system to correctly align the remote and local operators' environments. Both the abstract metaphors (e.g., shapes, arrows, etc.) and the avatar can be used in two different ways: (i) for pinpointing a specific object of interest and/or (ii) for showing how the objects should be manipulated by the local user. Generic virtual shapes or 3D arrows can be placed at the object's location to express the pinpoint action or the avatar itself can virtually point its arm towards the object of interest. The manipulation of the real objects can be expressed by an animated version of the corresponding 3D assets or by the avatar itself that shows how the objects should be manipulated by the trainee. In the proposed work, a set of pre-defined animations has been employed to present the virtual avatar movements; this choice is due both to guarantee the same type of visualization to all the trainees and to the lack of a real-time tracking system to measure the trainer's movements. The system allows the trainer to visualize both the virtual representation of the trainee and of the objects involved in the training procedure. The positions of the real tools is considered previously known whereas the position of the virtual trainer is constantly updated.

2.5.2 The System Architecture

Figure 2.21 illustrates the system architecture of SE. The trainer interacts in SE using an Oculus Rift DK2 Kit and a Microsoft XBOX 360 gamepad. Specifically, the Oculus Rift provides the trainer an immersive view of SE, whereas the gamepad allows the trainer to move and interact in SE. The trainee device is represented by the Microsoft HoloLens[13] glasses. Hence, the trainee can visualize the 3D virtual assets keeping his/her hands free to perform any possible task. In this experimental prototype, both the Oculus and the HoloLens have been connected to the same LAN. However, the same architecture can be generalized considering the Internet infrastructure. SE has been developed using the Unity3D game engine. The Oculus Rift DK2 is connected to a PC that acts as a server, whereas the HoloLens device

[13]https://www.microsoft.com/it-it/hololens

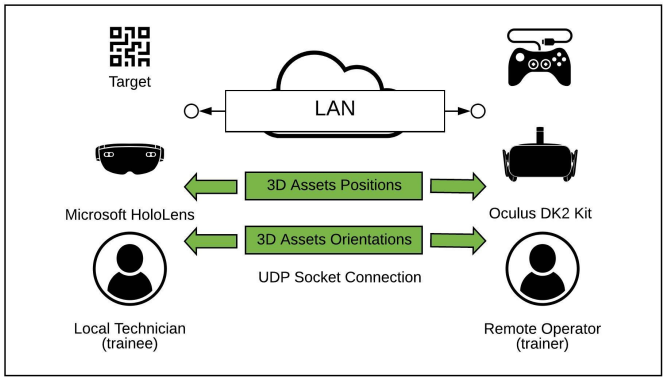


Figure 2.21: Left-side: the trainee environment. Right-side: the trainer environment. For evaluation purposes, they are connected on the same LAN.

acts as a client. Some libraries and APIs have been employed to manage several aspects of the application. The most relevant are:

- the SteamVR Plugin[14] that provides access to the Oculus Rift DK2 hardware;
- the Unet Unity API[15], to manage the multi-users architecture (specifically the High Level API);
- the Vuforia[16] library to track an image target allowing the SE alignment.

2.5.3 The Use Case

To compare the abstract and avatar interfaces, a robotic training task has been chosen. It consists of assembling the T42 3D printed hand [318], developed by the Yale School of Engineering and Science (the files are freely available to download[17]). The hand pieces list and the procedure for assembling it can be found online[18]. Although the T42 hand consists of a simplified version of a real manipulator hand, it is certainly related to the robotic arm area and thus it can be reasonably used to train a robotic technician. Its quite simple design and the use of non-hazardous

[14]https://assetstore.unity.com/packages/templates/systems/steamvr-plugin-32647

[15]https://docs.unity3d.com/Manual/UNet.html

[16]https://developer.vuforia.com/downloads/sdk

[17]https://github.com/grablab/openhand-hardware/tree/master/model%20t42

[18]https://www.eng.yale.edu/grablab/openhand/model%20t42/Fabrication%20-%20Model%20T42%201.0.pdf

materials ensure that it can be employed and tested by inexpert users, not trained for real industrial procedures.

Only a subset of the real hand pieces has been used in this training scenario. The goal of the trainee is to assemble the hand following the trainer's remote instructions. To complete the procedure, the hand has to be placed on a custom 3D printed flange attached to the terminal part of a real industrial robot. The real hand pieces have been placed at some predefined positions with respect to the image target (Fig. 2.22), allowing to correctly align their virtual counterparts.

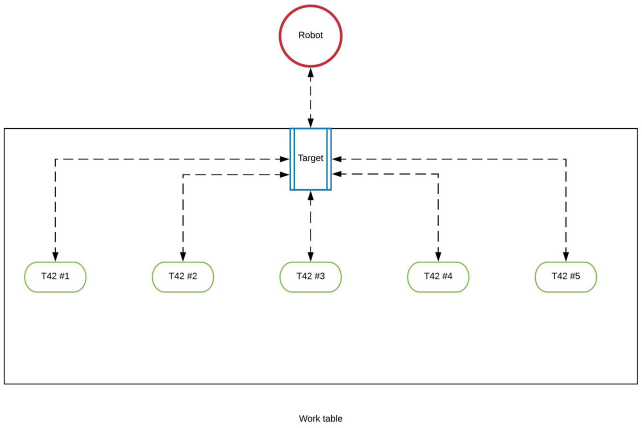


Figure 2.22: The image target is colored in blue. The real pieces and robot have been placed at some fixed positions with respect to the target.

2.5.4 The Interfaces

Both the trainee and the trainer can visualize each other in real-time. When the trainer moves the virtual camera in the virtual environment, the same motion is conveyed to a 3D avatar placed in the AR scenario. The same strategy can be adopted for the trainee's movements: when the trainee frames the image target, the related pose data are used to position the trainee avatar in the trainer's scenario.

In the next sections the AR and VR interfaces are presented and discussed. It is worth noticing that although the VR interface is briefly introduced, its effectiveness will not be evaluated. The focus will be on how the trainee perceives the remote instructions. Moreover, there are several works related to the use of immersive VR interfaces for maintenance operations and interested readers can find more details in [292, 257, 267, 102, 159].

The AR Interfaces

The two different AR interfaces (Fig. 2.23) differ only for some specific 3D assets, the abstract metaphors and the 3D avatar. Table 2.7 summarizes both interfaces.

Abstract Metaphors	Virtual Avatar
3D Arrow	Avatar VR
3D Cursors	3D Cursors
3D Hand Pieces	3D Hand Pieces

Table 2.7: The virtual assets of both interfaces.

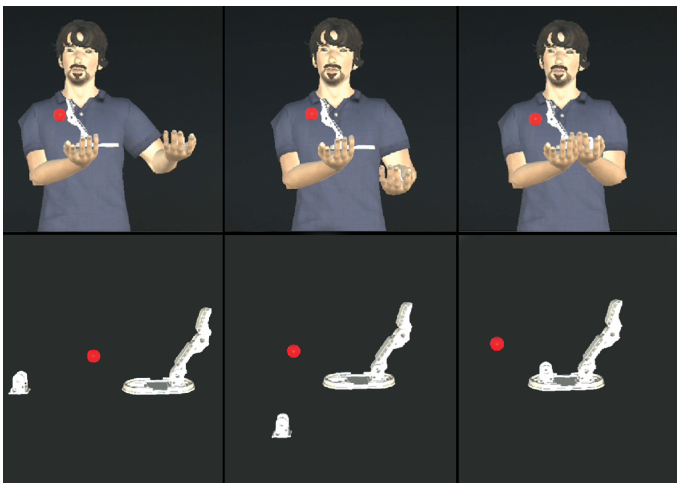


Figure 2.23: First row: the avatar assembly animations. Second row: the same animation done using the virtual hand pieces.

A virtual ray is casted from the user's head position following the head's viewing direction. When the ray hits a virtual objects, the spheres are rendered at the related coordinates. The virtual cursors are represented by small 3D red spheres. A virtual ray is casted from the user's head position following the head's viewing direction. When the ray hits a virtual object, the sphere is rendered at the related coordinates. The 3D arrows represent instead the virtual abstract metaphors. When the trainer pinpoints a specific 3D asset, a virtual arrow is positioned at the pointed position. The virtual avatar consists of a human worker virtual representation. To supply an effective assistance, some animations have been added both to the virtual T42 hand pieces and to the virtual avatar, thus allowing a fair comparison of both interfaces. The virtual abstract animations show how to correctly combine the real hand pieces, whereas in the other interface it is the avatar itself that shows to the

trainee how to correctly combine the real pieces (Fig. 2.23). Moreover, to foster the realism of the virtual avatar, three other animations have been added to the 3D avatar: (i) idle, (ii) walking and (iii) hand pointing animations.

Pre-defined animations have been used because the assembly procedure is made up of pre-determined steps. Moreover, they ensure that the visualization of the avatar animations is always the same, allowing to fairly analyze the effectiveness of the AR interface. The animations are played when the trainer presses the dedicated controller buttons, but for the avatar pinpointing action, which is applied in two different steps. Firstly, as the trainer presses the pinpointing button, a check on the ray-cast is performed to verify whether he/she is gazing to a virtual object. In case a collision is detected, the collision coordinates are used to apply an inverse kinematic algorithm on the right arm of the avatar, allowing to move the arm in the correct direction.

The VR Interface

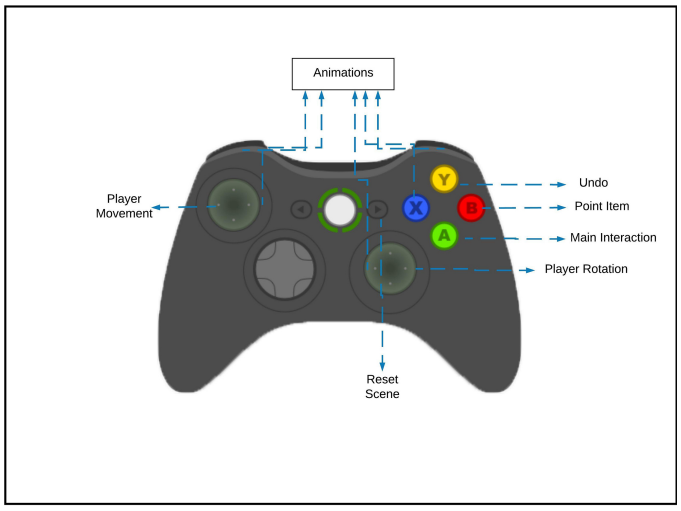


Figure 2.24: The trainer mapping input.

As it is possible to notice from Fig. 2.24, the left and right analog sticks are used to change the pose of the virtual camera, whereas the B button can be used to pinpoint the virtual models. Since the Oculus DK2 is a 6-DOF device, the trainer can look around the environment in all the possible directions, allowing to use the gaze as selection mechanism. The trainer can interact with a consistent number of virtual models. Since part of them are shared with the trainee, only the remaining ones are detailed in this section. The virtual robot used to hang the robotic hand is represented by a collaborative manipulator. The trainee is represented as a HoloLens virtual model whose movements are updated using the position and

orientation data of the real device. In addition, the trainer can visualize where the trainee is gazing, allowing him/her to have an idea of the trainee's intentions. Figure 2.25 shows the AR avatar interface and the VR one.

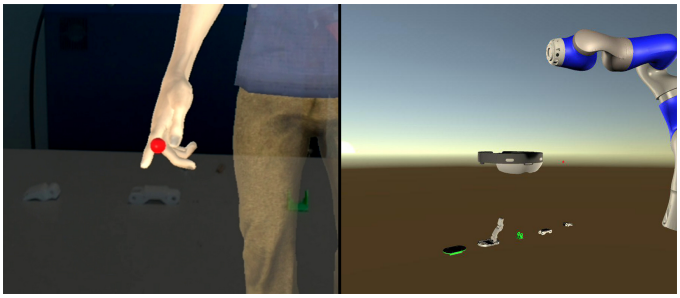


Figure 2.25: Left-side: the AR interface. Right-side: the same scene viewed from the VR interface.

2.5.5 The Interaction System

The trainee's operative zone has been divided into three areas: (i) the real objects' area (ROA). (ii) the working area (WA) and (iii) the assembled area (AA) (Fig. 2.26A). The animations of the abstract metaphors appear in front of the user in the "animation area" (ANA) (Fig. 2.26B). Only the animation representing the final step of the procedure behaves differently, because it is played at the end-effector position of the real manipulator. On the contrary, the virtual avatar animations are played by the character itself at its current position.

In Fig. 2.26C the workflow interaction is represented. As the trainer selects one of the virtual models, the corresponding real hand piece is highlighted in the trainee's ROA. If the trainee gives a positive audio feedback to the trainer, the trainer moves the selected 3D asset to the WA, allowing the trainee to verify whether the picked hand piece is the right one or not. Then, the trainer plays the corresponding assembly animation in the ANA. Finally, once the trainee confirms the completion of the procedure step, the virtual assembled piece is rendered in AA.

2.5.6 Tests and Results

In the following sections, tests and results are introduced and discussed.

Tests

Some tests have been carried out at Politecnico di Torino to compare the different training modalities. Twenty students have been identified, with ages between 20

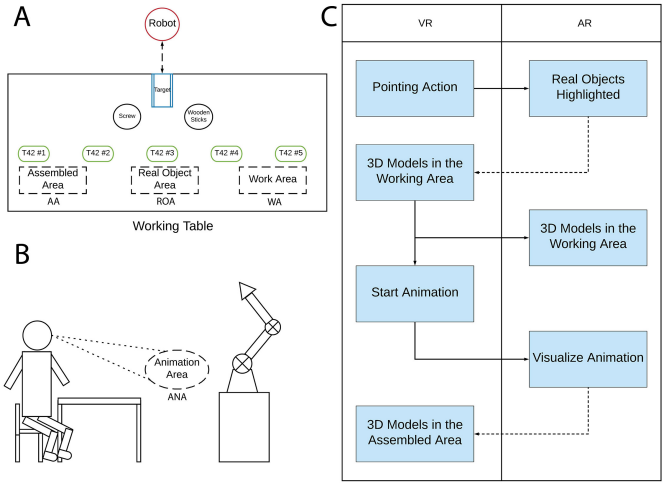


Figure 2.26: A) the ROA, WA and AA zones. B) the ANA zone. C) the work-flow interaction scheme.

and 28 years. The users had to build the T42 hand following the remote trainer's instructions. Since the comparison is assessed only from the AR point of view, the figure of the trainer has been interpreted by one of the paper's author. Two different tester groups (called A and B) have been organized: the A group evaluated the abstract metaphors interface, whereas the B group assessed the virtual avatar-based interface. Tests have been done following the subsequent procedure:

1. users have been introduced to the test, explaining that a remote operator would have guided them during the assembly of the T42 hand;
2. users of both groups have tested the corresponding interface;
3. after the test, a questionnaire has been submitted to both groups.

Two questionnaires have been prepared (QA and QB), one for each group. Both QA and QB are divided in three different sections: the first one concerned general information about the user and his/her knowledge of AR whereas the second section was composed by nine statements (5-point Likert scale) taken from [347]. The third section was composed by eleven statements (5-point Likert scale) concerning the clearness and effectiveness of the abstract metaphors (for QA) or of the avatar (for QB). Moreover, it focused on analyzing whether the visualization of the avatar could improve or not the sense of human-human collaboration. The questionnaire also provided an open text form for free comments.

Figure 2.27A shows the starting configuration of the real hand pieces. One of the two fingers was already assembled and inserted into the base (Fig. 2.27B). It could be used as a reference model. Hence, the users had to assembly the other

finger, plugging it into the base. Some additional tools have been provided to the testers to complete the training procedure: two tiny wooden sticks, two screws and two bolts. The real hand pieces and the real manipulator were placed at some predefined positions with respect to the target. The users started the procedure by sitting down on a chair positioned in front of a table. Then, after having assembled the robot hand, they had to plug it on a 3D printed support attached on the industrial manipulator end-effector (Fig. 2.27D).

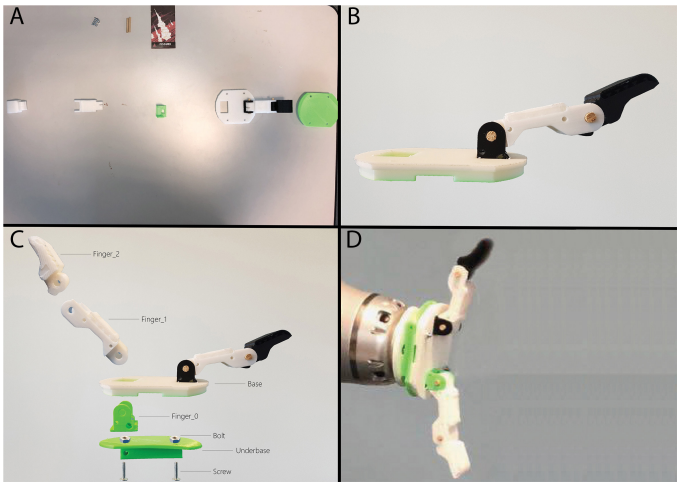


Figure 2.27: A) the starting pieces' configuration. B) the assembled finger used as reference. C) the hand pieces names convention. D) the assembled hand attached to the real robot end-effector.

The entire procedure was composed of seven different steps (refer to Fig. 2.27C for the hand pieces name convention):

1. take the Finger_0 and the Base;
2. plug the Finger_0 into the Base (creating a new piece called F_Base);
3. take the Finger_1 and the Finger_2;
4. combine the Finger_1 and the Finger_2, using the wooden sticks (creating a new piece called Finger);
5. take the F_Base and combine it with the Finger, using the wooden sticks (crating a new piece called Hand_1);
6. take the Underbase and attach it to the Hand_1, using the two screws and the two bolts (creating a new piece called Hand_2);
7. plug the Hand_2 on the robot's end-effector.

The feedback audio channel has been established using two smartphones and two Bluetooth earphones. To ensure that the same instructions were conveyed in the same way to the users, a text file has been prepared with the instructions that the trainer had to provide to trainee for each step of the procedure. Figure 2.28 shows some testers performing the training task.

Figure 2.28: Four users performing the training procedure.

Results

Table 2.8 shows the results of the first and second questionnaire sections. Despite 90% of the users stated to know the AR technology, just over the half of the participants had experience with an AR application. Furthermore, all the users declared that they did not have any experience with the T42 robotic hand. A two-tailed t-test ($p = 0.05$) with unequal variance has been performed on the data of the second and third sections of the questionnaire and no statistically significance differences have been found between the two groups. However, it is possible to make a preliminary discussion using the mean (M) and standard-deviation (SD) data (effect sizes $d_{A1} = 0.35$, $d_{A2} = 0$, $d_{A3} = 0.16$, $d_{A4} = 0.67$, $d_{A5} = 0.93$, $d_{A6} = 0.38$, $d_{A7} = 0.5$, $d_{A8} = 0.29$, $d_{A9} = 0.39$, $d_{Q1-Q12} = 0.58$, $d_{Q2-Q13} = 0$, $d_{Q3-Q14} = 0.62$, $d_{Q4-Q15} = 0.59$, $d_{Q5-Q16} = 0.29$, $d_{Q6-Q17} = 0.14$, $d_{Q7-Q18} = 0.19$, $d_{Q8-Q19} = 0.51$, $d_{Q9-Q20} = 0.50$, $d_{Q10-Q21} = 0.35$, $d_{Q11-Q22} = 0.08$). The abstract metaphor data are generally higher and less distributed than the avatar one (but for the A2 statement that was negative worded). It seems that the virtual arrows have been considered more efficient than the avatar one to clearly explain the steps of the procedure. Moreover, the outcomes suggest that the abstract metaphors have eased the assembly procedure learning process (probably also the use of the

#	**Questions**		
1	Age	Average = 24.5	
2	Gender	70% Male	30% Female
		YES (%)	**NO (%)**
3	Do you know what Augmented Reality is?	90	10
4	Have you ever used an Augmented Reality application?	65	35
5	Do you know the Yale Model T42 Robotic hand?	0	100
6	Have you ever assembled the Yale Model T42 Robotic hand?	0	100
		Metaphors	**Avatar**
		AVG-SD-M-IQR	**AVG-SD-M-IQR**
A1	I think the system was easy to use	4.4-0.51-4-1	4.2-0.6-4-0.75
A2	I would need the support of a technical person to be able to use this system.	1.7-1.06-1-1	1.7-0.09-1.5-1
A3	The user interface of this system is pleasant.	3.6-0.5-4-1	3.7-0.67-4-1
A4	I can effectively complete my tasks using this system.	5-0-5-0	4.8-0.42-5-0
A5	This system gives me clear instructions.	4.9-0.31-5-0	4.5-0.5-4.5-1
A6	It was easy to learn how to use this system.	4.9-0.31-5-0	4.7-0.67-5-0
A7	I would recommend this system to my friends or colleagues.	4.7-0.48-5-0.75	4.4-0.69-4.5-1
A8	The feedback given by this system is easy to understand.	4.3-0.67-4-1	4.5-0.7-5-1
A9	Overall, I am satisfied with this system.	4.3-0.48-4-0.75	4.5-0.52-4.5-1

Table 2.8: The results of the first two questionnaire sections (AVG, SD, M and IQR are the average value, the standard deviation, the median value and the interquartile range, respectively).

AR interface itself). Despite these outcomes, overall the users slightly preferred the avatar interface (A9). This result seems also confirmed by the A8 outcomes. Probably, since human beings are used to see human figures during their everyday life, the visualization of a human form simplified the understanding of the interface. Concerning the third section of the questionnaire, Table 2.9 shows the abstract metaphor and avatar results. The outcomes have been aggregated computing the M and SD values. Statements Q4/Q15, Q6/Q17 and Q9/20 were negative worded.

	Metaph. (M) - (SD)		**Avatar (M) - (SD)**
Q1	4.4 - 0.69	**Q12**	4 - 0.66
Q2	4.6 - 0.69	**Q13**	4.6 - 0.15
Q3	4.6 - 0.69	**Q14**	4 - 1.15
Q4	1.4 - 0.69	**Q15**	2 - 1.33
Q5	3.9 - 1.19	**Q16**	3.5 - 1.51
Q6	1.9 - 1.28	**Q17**	2.1 - 1.45
Q7	4.6 - 0.69	**Q18**	4.4 - 1.26
Q8	4.8 - 0.41	**Q19**	4.5 - 0.71
Q9	1.1 - 0.31	**Q20**	1.5 - 0.08
Q10	3.7 - 1.15	**Q21**	4.1 - 1.10
Q11	2.9 - 1.10	**Q22**	2.8 - 1.13

Table 2.9: The results of the third questionnaire section (M represents the average value and SD the standard deviation). See Appendix C.3 for the complete questionnaire.

Also in this case no statistically significant difference have been found. Overall, the results seem to confirm the ones found in the second section, suggesting that the avatar did not improve the sense of human-to-human collaboration.

Although it was not possible to clearly verify any statistical difference between the interfaces, it could be deduced that the interface less "resource" demanding should be preferred. Designing an AR avatar interface requires great effort to produce high-realistic human models and animations. In case of real-time animations, the computational cost and the related resources may increase considerably. Moreover, considering that it is extremely difficult to visualize an entire human body using the current wearable AR devices (the HoloLens FoV is around 35 °), the users could only see the terminal part of the arm, reducing the sense of human presence. Moreover, the relative huge size of the virtual avatar may have generated occlusion problems, overlapping the virtual objects onto the real ones and straining the users' sight. Taking into account the obtained results and the above considerations, it seems that the abstract metaphor-based interface should be preferred for managing remote maintenance operations.

However, it seems also that the audio channel played a key-role during the collaboration (Q10/Q11/Q21/Q22). This outcome seems to be verified from the analysis of the current state of the art related to the AR remote assistance systems. In fact, a remote assistance system is normally made up by both audio and video communication channels. Hence, it becomes important to verify which is the impact of the audio on the effectiveness of both interfaces and on the sense of human presence. Therefore, an additional test has been done and it will be introduced in

the next section.

2.5.7 Additional Tests

	Metaphors		Avatar				Metaphors		Avatar	
	M	**SD**	**M**	**SD**			**M**	**SD**	**M**	**SD**
A1	4.6	0.57	4	1		**Q1/Q12**	4.6	0.57	3.6	1.15
A2	1	0	2	1		**Q2/Q13**	5	0	3.3	1.52
A3	4.3	0.57	2.6	0.57		**Q3/Q14**	3.3	1.52	4.6	0.57
A4	5	0	4.6	0.57		**Q4/Q15**	2	1	1.33	0.57
A5	4.6	0.57	4.3	1.15		**Q5/Q16**	2.66	1.52	4	0
A6	5	0	4.3	0.57		**Q6/Q17**	2.66	1.52	1.6	0.57
A7	4.6	0.57	4.3	0.57		**Q7/Q18**	4.3	1.15	4	1
A8	4.6	0.57	4.3	1.15		**Q8/Q19**	5	0	3.66	1.52
A9	5	0	4	1		**Q9/Q20**	1	0	2	1.73

Table 2.10: Additional test results.

An additional evaluation has been done to verify whether the audio channel has lowered the differences between the abstract and avatar interfaces. The same training procedure has been used (Sec. 2.5.6), but with a different feedback strategy based on a wizard mechanism: the users could only inform an external collaborator if they had figured out the procedure by saying “Yes/No” and they could only ask to repeat a specific step of the procedure. No other comments or dialogues were allowed.

Six new volunteers, divided in A and B groups, have been involved into the additional tests. The user’s average age was equal to 21 and, as for the previous test, they had some experience with AR applications but not with the Niryo robot arm. The remaining outcomes are shown in Table 2.10. A two-tailed t-test ($p = 0.05$) with unequal variance has been performed on the data of the second and third sections of the questionnaire and, as for the previous test, no statistically significance differences have been found between the two groups. However, it is possible to make a preliminary discussion using the mean (M) and standard-deviation (SD) data (effect sizes $d_{A1} = 0.81$, $d_{A2} = 1.41$, $d_{A3} = 2.88$, $d_{A4} = 0.81$, $d_{A5} = 0.36$, $d_{A6} = 1.63$, $d_{A7} = 0.57$, $d_{A8} = 0.36$, $d_{A9} = 1.41$, $d_{Q1-Q12} = 1.09$, $d_{Q2-Q13} = 1.54$, $d_{Q3-Q14} = 1.15$, $d_{Q4-Q15} = 0.81$, $d_{Q5-Q16} = 1.23$, $d_{Q6-Q17} = 0.86$, $d_{Q7-Q18} = 0.30$, $d_{Q8-Q19} = 1.23$, $d_{Q9-Q20} = 0.80$).

Also in this case, the abstract AR interface has been deemed more suitable than the avatar one for most part of the questions. Moreover, it has been assessed more gratifying than the avatar interface. The virtual arrows were found to be more useful to indicate the real hand objects and more effective to explain the assembly procedure. The results concerning the “sense of presence" (Q5/Q16 and Q3/Q14)

indicate an opposite trend. In fact, in no audio condition the virtual avatar has been considered more suitable to express the presence of the trainer in the trainee environment. It can be inferred that the combination of no audio conditions and abstract metaphors may make the shared environment less "human" collaborative than the no audio and avatar interface. However, although the virtual avatar seems to improve the sense of human-to-human collaboration with no audio condition, the users have deemed more effective the abstract metaphors. Hence, it becomes necessary to investigate whether the sense of human presence is unnecessary in industrial scenarios. Moreover, it is equally important to realize whether the avatar could be effectively used in scenarios that require more complex physical human gestures, analyzing the interactions in both audio and no audio conditions.

2.5.8 Conclusions

The proposed comparison aimed at investigating whether a virtual human agent could improve the efficiency and the sense of human-to-human collaboration during a training robotic assembly procedure. The presented system is composed of a shared environment which allows two operators to interact using two distinct AR and VR interfaces. Specifically, a local technician, wearing a wearable AR device, can receive instructions from a remote user operating in an immersive VR environment. Two distinct AR interfaces have been compared: the first one is made up by abstract metaphors whereas the second one is characterized by the presence of a virtual avatar. The preliminary results indicate that in an industrial scenario, it should be employed the interface that requires less resources to be managed. Since the designing process of the abstract metaphors is less compelling than the avatar one, it may be preferred for these types of scenarios. Furthermore, the current technological limitations impose the deployment of small virtual assets that can be easily visualized using wearable AR devices. Another important outcome is that the audio channel seems to play a fundamental role and additional tests should be done to statistically verify whether the audio instructions can completely replace any form of graphical assets. However, in no audio condition, the virtual avatar seems to foster the sense of human-to-human collaboration. Although at the current state, it seems that the sense of presence of the remote trainer is substantially necessary in a industrial scenario, additional tests should be done to evaluate the relation among the audio channel, the avatar and the task itself.

2.6 VR in Telerobotics

Robot manipulators were traditionally employed to eliminate human effort in elementary and repetitive tasks, improving process efficiency and reducing faults.

However, many complex applications still require a crucial human level of intelligence and perception. In such cases, the manipulator can be configured to serve as an extension of the human operators, providing them control and sensory feedback. In addition, such teleoperation systems can allow the human operator to remotely interact with both the robot and its surrounding environment.

Teleoperated robot systems have shown their versatility in a plethora of industrial, healthcare and commercial scenarios. They are crucial for hazardous operations or inaccessible environments, such as disaster response [214] or space station maintenance [9]. The robot teleoperation is also becoming increasingly relevant in the healthcare sector, reducing the risk of infectious disease transmission [436]. Even before the Covid-19 pandemic, extensive resources have been invested into remote surgery [424, 446, 156], which requires a highly skilled operator and extensive training. In this case, the teleoperation saves travel time and allows the skilled users to remotely apply their knowledge in time-critical tasks that may save lives. A similar highly skill-dependent task is the remote welding operation [263], where the manipulator has to follow the human motion with high levels of accuracy. Both applications require an accurate representation of the manipulator's environment and fine control of the robot's end-effector. Basic teleoperation interfaces usually provide an RGB video stream of the robot's surroundings, with possible sound or force feedback. However, the lack of depth in this data often makes it hard to precisely control the robot end-effector pose. To compensate for this, a considerable amount of works have studied and analzyed the effectiveness of the immersive VR interfaces for controlling and collaborating with robotic manipulators. Gammieri et al. [134] presented a VR scenario to foster the human-robot collaboration, effectively coupling the virtual environment with a real manipulator. They also introduced a virtual interface that provides users the ability to control the manipulator using both direct and inverse kinematics. An analysis of human reactions in a human-robot collaboration context is carried out using an immersive VR system in [144]. One interesting result shows that there may be a correlation between a user's reaction to a robot manipulator and his/her previous experience with VR. Holubek at al. [183] proposed an immersive VR interface to program a robotic arm. The real environment is firstly modeled using a CAD software and then the users can interact with the virtual robot using the dedicated VR device. Regarding the virtualization of the real robot environment, works in [339, 451] employed RGB-D cameras to pre-scan the real surrounding, visualizing it with an immersive VR device. Perez et al. [339] presented a VR interface that facilitates human workers' training in an industrial robotic cell, whereas in [451] the authors compared a pre-scanned environment (*full-information*) with one made up by only a virtual robot (*preprocessed*) in a path-following task. The outcomes show that using in the full-information interface, the users were faster than with the preprocessed one and no difference has been detected in the robot accuracy for the considered task. In

[282], the authors analyzed the effectiveness of several types of interfaces (immersive, speech, gestures, and combination of them) for controlling a hyper-redundant robot. The comparison of those interfaces with 2D conventional ones show that the immersive ones improved visual feedback, situational awareness and control accuracy.

Although the VR interfaces have proved to give satisfactory results for operators' training, simulation of industrial robotic cells and robot control, they are limited to pre-scanned or simulated environments that do not represent in real-time the real robotic cell. On the contrary, by using RGB-D sensors, the robotic cell can be captured and reconstructed in real-time at the operator side. In this context, such systems will be referred with the term "Enhanced Virtual Reality" (EVR) systems. Kohn et al. [229] presented an object recognition system to lower the amount of digital data required to represent the real scenario. The RGB-D cameras are used to recognize and detect objects placed in the robotic cell which are then replaced by virtual meshes and the related data are removed from the streaming, thus reducing its size. Pick-and-place tasks have been also evaluated using the EVR interfaces [480, 125, 479]. In [480] and its improved version [479], the authors compared an immersive an EVR interface with different interaction paradigms (kinesthetic approach, offline programming, and positional tracker with monitor) to control a robotic manipulator. The main results show that even though the users were faster using the kinesthetic approach than the others, the immersive interface was the most appreciated in terms of usability and likability (similar results have also been reported in [305]). A comparison between a tracked motion controller (Oculus Touch) and a fixed 6-axis controller in an EVR based teleoperation task is proposed in [125]. The outcomes show that the Oculus Touch allows faster performance due to increased speed in the movement planning. Further works can be found in [421], which proposes two different robot controlling algorithms for an EVR interface or in [139, 505], where authors employed deep learning approaches to map the operator's movements to the robot joint's angle and to reconstruct the real objects manipulated by the robot, respectively.

Analyzing the state of the art, there is a lack of studies that evaluate the effectiveness of the EVR interfaces for robot arm teleoperation tasks that require high levels of accuracy. Hence, a novel streaming strategy developed for this Ph.D dissertation to transfer high resolution point clouds at high frame rate is presented and detailed. Furthermore, a series of experiments to control the robot end-effector in path following scenarios are discussed by analyzing both objective and subjective parameters. Finally, a discussion related to the impact of point cloud quality on the teleoperation itself is proposed.

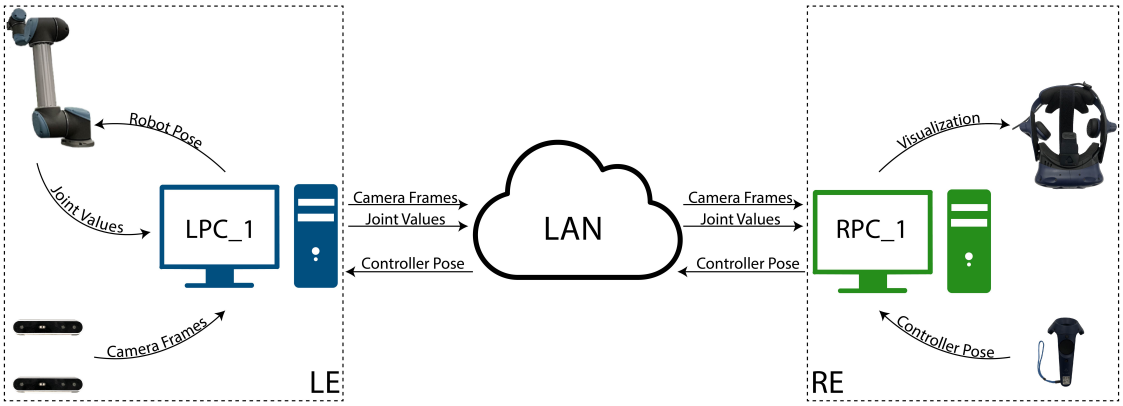


Figure 2.29: Left-side: LE with the robot and the depth cameras. Right-side: RE with the VR device.

2.6.1 The Hardware and Software Architectures

The proposed framework consists of two different environments (Fig. 2.29): the Local and Remote Environments (LE and RE, respectively). LE is characterized by a Universal UR5 robot arm, a PC running Ubuntu 18.04 with ROS Melodic and two Intel RealSense D415 cameras. RE represents the user environment and it includes a PC running Windows 10 and the HTC Vive Pro along with one controller. LE and RE are connected on the same LAN and they exchange data using both Transmission Control Protocol (TCP) and UDP protocols. Regarding the software architecture, the camera frames are sent using a custom solution based on the TCP/UDP protocols, whereas the robot data is exchanged between an application developed in Unity3D and the ROS controller on the LAN using the ROS# Library[19]. Since the quality of the visualization of the point cloud is extremely important for an effective functioning of the framework, the details of its streaming and rendering are discussed in the next section.

2.6.2 The Point Cloud Streaming and Rendering

A 3D point cloud is usually represented as *"a set of points* $\{P_i\}_{i=1}^{n}$*, embedded in the 3D space and carrying both geometry and attribute information"* [52], that is, each point carries both color and depth information along with other attributes. The methodology developed for the proposed EVR interface handles the color and depth data separately and is composed by three different steps: (i) the Camera Handshake, (ii) the Streaming and (iii) the Rendering.

[19] https://github.com/siemens/ros-sharp

Camera Handshake

During this step, RE is notified regarding the camera configuration (number of cameras and camera parameters) of LE. RE, at the software level, is composed of two different applications (Fig. 2.30). The C++ server is responsible for the camera streaming and it forwards the camera frames to the Unity3D application that is used for rendering the remote scene. Firstly, data regarding the number of cameras, their resolutions and intrinsic parameters are sent to the C++ server over a TCP connection (Fig. 2.30 (1)). Then, these data are forwarded to the Unity3D application over a TCP localhost connection. Hence, both applications know the LE camera configuration and they can allocate the related data structures to handle the streaming and rendering of the scene. Once the Unity3D application has allocated the required data structures, it sends back to LE an acknowledgment (Fig. 2.30 (2)), starting the real streaming.

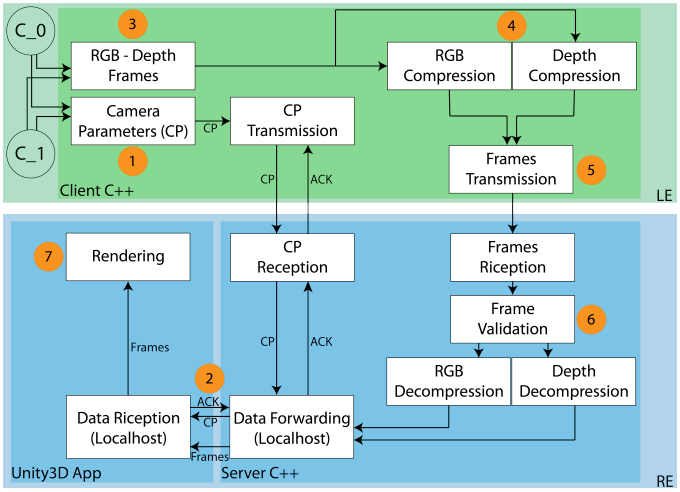


Figure 2.30: The point cloud streaming. The image frames captured by cameras C_0 and C_1 are streamed over the network to RE. After a validation process, they are decompressed and rendered in the Unity3D application.

Streaming

The C++ client in LE sends the camera frames to the C++ server over an UDP connection (Fig 2.30 (3)). The UDP protocol has been chosen as it is usually employed for multimedia streaming and it ensures high speed transmission. Then, the frames are forwarded over the TCP localhost to the Unity3D application to render the remote scene. Since a point cloud carries a considerable amount of data (e.g., with 2 bytes for each depth value, 3 bytes for each color value, and a 1280x720 camera resolution, each frame carries about 4.6 MB of raw data) and UDP does not guarantee data transmission, a compression/decompression step and a frames

Table 2.11: The compressed frames. Each line represents a different compressed frame. The compression ratio is on average 9:1.

Compr. Color (byte)	Compr. Depth (byte)	Tot Compr. (byte)	Ratio Raw/Compr.
44967	466668	511635	9.006
44786	468980	513766	8.969
44761	466548	511309	9.012

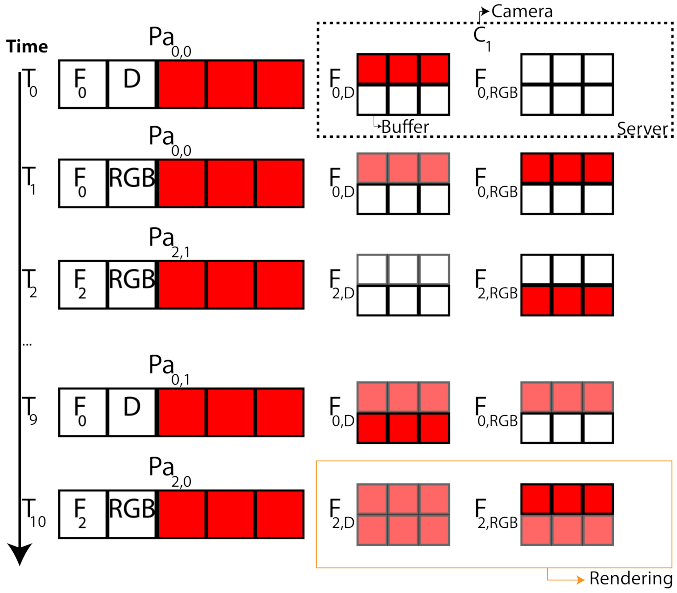


Figure 2.31: The frames' validation procedure. As time passes, the frame buffers are randomly filled and only the frames that are full received (color and depth) are rendered, discarding the previous ones.

validation check have been developed to guarantee an effective data transmission at maximum frame rate. Specifically, the JPEG compression has been applied to the color frames using the libjpeg-turbo library[20], whereas the depth frames have been compressed using the approach proposed in [482] (Fig. 2.30 (4)). This methodology ensures high compression ratios (Table 2.11) allowing to send multiple camera frames over a standard 1 Gigabit Ethernet cable at maximum resolution (1280x720) and frame rate (30 fps).

After the compression, each frame (both color and depth) is divided in UDP packets of maximum 1500 bytes that are sent to LE using the Asio library[21]

[20] https://www.libjpeg-turbo.org/About/TurboJPEG

[21] http://think-async.com/Asio/

(Fig. 2.30 (5)). In order to guarantee the correct frame reconstruction, each packet carries a header containing: (i) the packet type (RGB or depth), (ii) the camera index, (iii) the frame index, (iv) the buffer index, and (v) the compressed frame size that is used to decompress the frame at the receiving side. The remaining bytes are used to carry the payload (the compressed frame data). Figure 2.31 depicts the frames validation check procedure (for the sake of clarity, only one camera is considered in this example but the proposed system supports multi-camera configuration). Let $Pa_{k,i}$ represent an i-th packet of frame k ($1 \leqslant i \leqslant L$, with L representing the number of frame packets and $0 \leqslant k \leqslant N$ with N representing the number of frames. Let C_n be the n-th camera, with $n \geqslant 1$ and $F_{k,type}$ be the k-th frame of type RGB or D (depth). At time T_0 a packet $Pa_{0,0}$ of frame $F_{0,D}$, camera C_1 is received by the server and it starts filling the corresponding buffer. Since UDP does not guarantee packet ordering, it is possible that at time T_1, the server receives another packet $Pa_{0,0}$ of the other frame $F_{0,RGB}$. As time goes by, the packets may arrive not in order and the frame buffers are filled in a non linear way (it is unlikely that the frames arrive ordered as they have been transmitted). Hence, it is not possible to determine in advance which frame will be fully collected, nor whether both color and depth frames can be entirely received. To overcome this limitation, only the frames that are fully received (color and depth) are considered for rendering and the previous frames are ignored and deleted. Referring to Fig. 2.31, at time T_{10} all the buffers of frame F_2 are full. Hence, they will be rendered and all previous frames are ignored. This procedure has been generalized for a multi-camera configuration and only the frames that are fully received from all the cameras are rendered. Therefore, it is ensured that distinct frames capturing the same scene from different viewpoints at the same time are rendered and visualized at the same moment. (Fig. 2.30 (6)).

Rendering

After the frames reconstruction, the frames are sent to the Unity3D application over the TCP localhost for the rendering stage (Fig 2.30 (7)). Let $D_{i,j}$ be the depth matrix, with i and j being two indices going from 0 to the matrix width and height, respectively. The depth scaled value $ds_{i,j}$ can then be computed as:

$$ds_{i,j} = D_{i,j}s, \tag{2.8}$$

where s is a scale factor. The coordinates of the 3D point are computed as:

$$\begin{cases} x = ds_{i,j}(i - p_x)/f_x \\ y = ds_{i,j}(j - p_y)/f_y \\ z = ds_{i,j}, \end{cases} \tag{2.9}$$

where f_x, f_y, p_x, p_y are the camera intrinsic parameters sent during the Camera Handshake step, corresponding to the focal length and the principal points, respectively. As this computation is done on the CPU, the vertices are sent to the GPU

to apply the color data.

2.6.3 The Proposed System

The immersive user interface, the robot controlling strategy and the camera calibration are presented and discussed in the next sections.

The User Interface

The users can visualize and interact with the remote robot environment by using the HTC Vive Pro. The immersive user interface provides two distinct types of interaction: teleporting and robot controlling. Using the former, the users can virtually teleport themselves everywhere in the virtual scenario by pressing the touchpad button of the Vive controller. The latter allows the users to remotely control the robot arm end-effector by pressing the side joystick button. When the side button is kept pressed, the joystick pose with respect to the robot base is sent to the ROS-based robot control scheme over the LAN. Furthermore, a virtual reference system has been added to the virtual joystick to help users visually tracking the end-effector translation and rotation.

Robot Teleoperation

The manipulator teleoperation is based on the Vive controller's pose with respect to the virtual robot reference system. The 6-DOF pose of the Vive joystick with respect to the robot end-effector is expressed as an SE(3) transformation matrix. As the teleoperation is enabled, the reference end-effector pose with respect to the robot base and the reference joystick pose with respect to the end-effector are stored. The goal end-effector offset with respect to the initial reference pose is computed at every control loop update. Hence, the robot end-effector follows the joystick pose with respect to arbitrary reference poses determined by the user. This ensures a consistent control interface from every point of view in the virtual environment.

Camera Calibration

To properly reconstruct the same scene from different points of view, the RGB-D cameras have been extrinsically calibrated using the procedure proposed in [21]. Moreover, to detect the robot pose, an Aruco marker [138] has been placed at a known position with respect to the manipulator. Once the marker is detected by one of the cameras, it is possible to derive the position and orientation of the robot arm in the virtual environment.

2.6.4 A Preliminary User Study

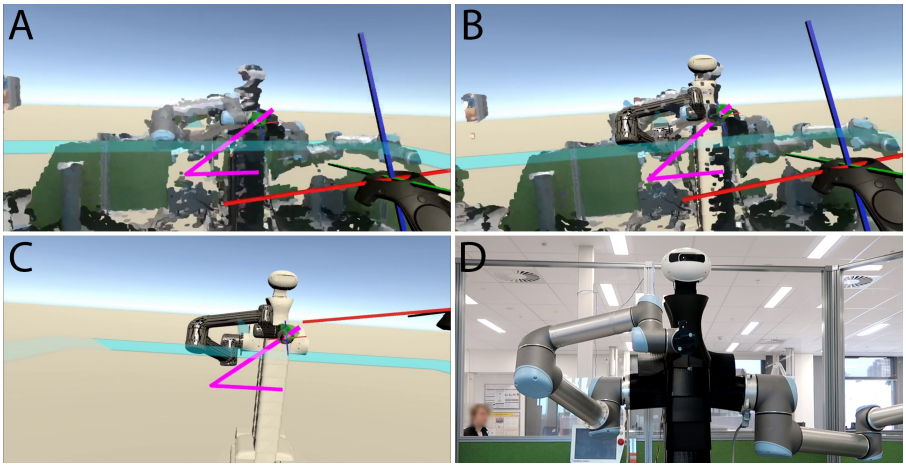


Figure 2.32: The evaluated teleoperation interfaces. Subfigure A) presents I_{EVR}, subfigure B) presents I_{EVRR}, subfigure C) presents I_{VR}, and subfigure D) the real robot.

The key goal is to evaluate the effectiveness of the proposed EVR interface to control a manipulator for highly accurate and precise tasks. To do so, a preliminary user study involving a limited number of users has been firstly done. Then, a more rigorous assessment has been carried out with a consistent number of users (Sec. 2.6.5).

Since the extrinsic and marker calibrations introduce inaccuracies, the proposed system (henceforth, I_{EVR}) has been compared with a "pure" VR version of the interface (I_{VR}) that does not require calibration. Furthermore, in order to properly assess whether the point cloud resolution is enough to clearly detect the robot end-effector in the virtual environment, I_{EVR} has been also compared with an EVR version that has the robot CAD model overlaid on the reconstructed one (I_{EVRR}). Figure 2.32 shows the evaluated interfaces and the real manipulator.

Six users (with ages between 25 and 31 years old) were asked to complete four distinct tasks using the above three interfaces. The users had to control the real robot end-effector (EE), doing two different types of task. In the Pose Task (PT), the users had to place the real robot EE in a specific position and orientation in the 3D environment. For the Speed Task (ST), the users had to move the real robot EE, following a pre-set trajectory matching a very specific velocity. The EE positions/orientations (for PT) and the EE movement along the trajectories (for ST) were prerecorded using the real manipulator. Hence, it has been possible to compare the motion of the EE controlled by the testers with a joint base-line, guaranteeing an objective assessment of the users' performance. Both objective and subjective parameters have been collected: i) the end-effector pose (PT), ii) the

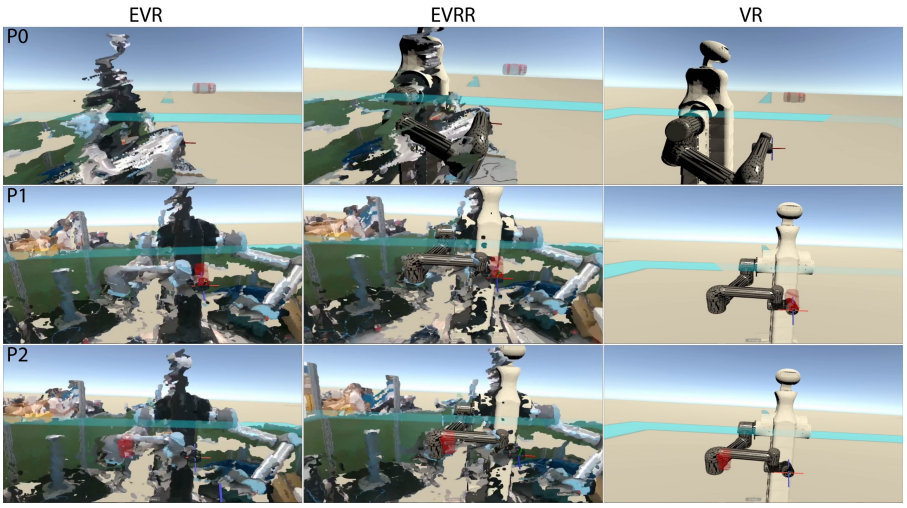


Figure 2.33: The pose tasks. The final position is highlighted by the red virtual Ghost.

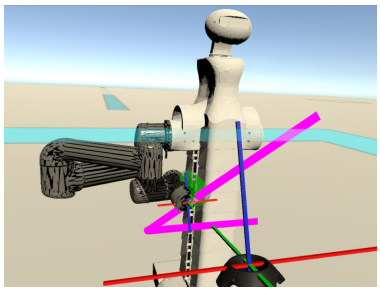
end-effector trajectory with respect to time (ST), iii) the usability based on the SUS questionnaire [48], and iv) the workload based on the NASA-TLX questionnaire [169].

The Pose Task

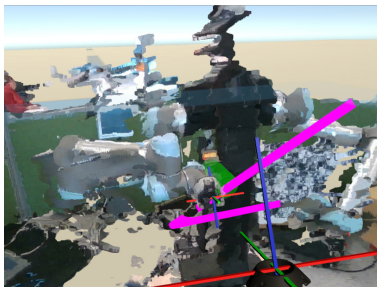
The users had to complete three tasks, positioning the real robot EE in some specific locations of the 3D space. A virtual asset (called "Ghost"), representing the robot EE, has been used to highlight the locations in the virtual space. Even though in PT the users are not forced to follow a specific trajectory, the tasks have been divided so as to carry out three different types of motions: pure translation (P0), pure rotation (P1) and roto-translation (P2) (Fig. 2.33). Starting from a fixed pose of the real robot EE (same for all the tasks), the users had to control the robot EE so as to place it "inside" the Ghost, matching as close as possible its positions and orientations. Each PT ended when the user was satisfied with his/her performance. Ideally, P0 required only translation, P1 only rotation and P2 both translation and rotation. Figure 2.33 shows the P0, P1 and P2 tasks.

The Speed Task

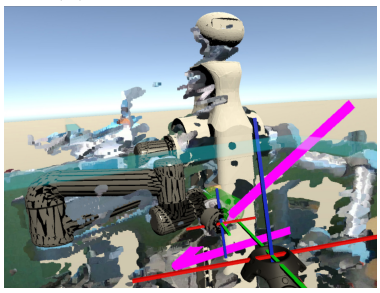
For the preliminary evaluation, ST is composed by only one task (S0). The users had to move the real robot EE along a pre-recorded trajectory, matching the pre-computed velocity. The trajectory is represented by a virtual purple line. The Ghost proceeds along the pre-recorded trajectory with the pre-determined real robot EE velocity and the users had to move the real EE so as to match as close



(a) The VR interface.



(b) The EVR interface.



(c) The EVRR interface.

(d) The real robot.

Figure 2.34: The users had to move the robot end-effector along the purple line, following the Ghost's movements.

as possible the Ghost position. In order to check whether the EE controlled by the users is accurately reproducing the Ghost motion, the collisions between the virtual collider robot EE and the one of the Ghost are at checked every frame. In case a collision is detected, the Ghost keeps moving along the trajectory, otherwise it stops moving, giving time to the users to align the real EE with the virtual one. The task ends when the real EE arrives at the last trajectory position. The pre-computed trajectory consists of a pure translational motion, without any rotations. Finally, the Ghost begins moving from a location far from the initial trajectory position to give the users time to get ready for the simulation. It switches its color to green when it crosses the purple line, highlighting the beginning of the task. Figure 2.34 depicts the S0 task.

Furthermore, a video showing the PT and ST tasks can be found at[22].

Preliminary Results

The one-way ANOVA test has been used to verify whether significant differences exit among the interfaces, showing p values greater than 0.05 ($p_{SUS} = 0.127$,

[22] https://youtu.be/qgY5OKUMrg0. Note that in the video the EVR, EVRR and VR interfaces are called MR_S, MRR_S and VR_S, respectively.

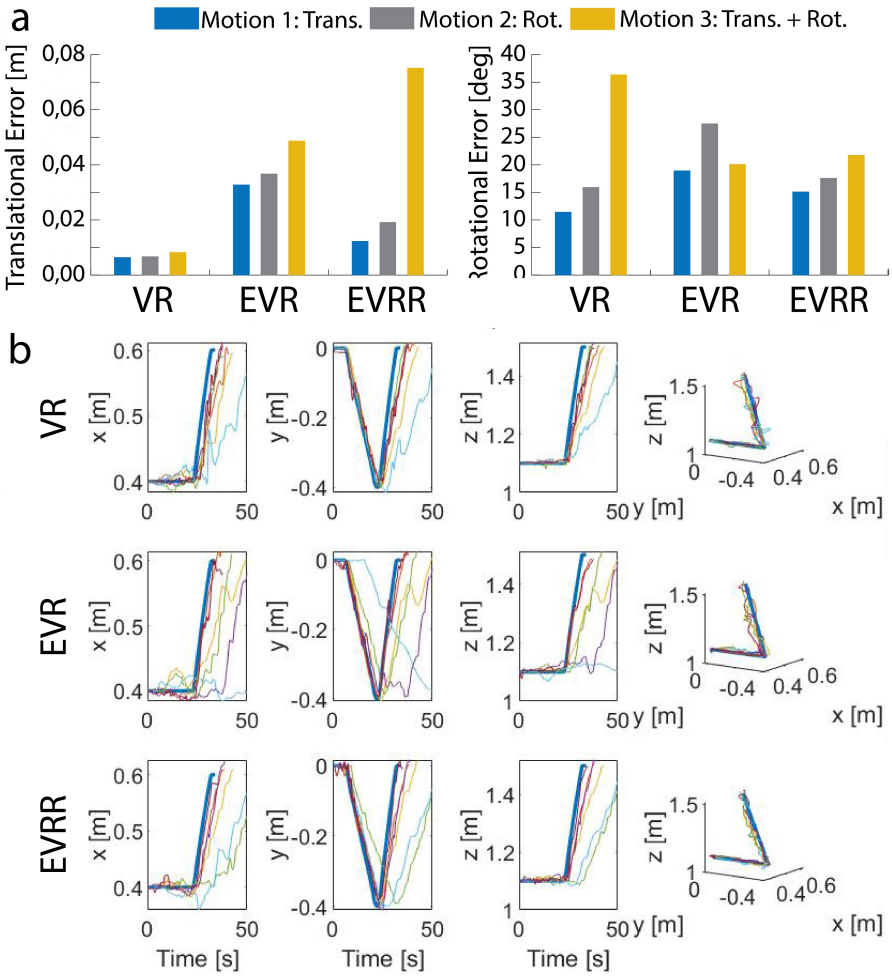


Figure 2.35: a) Left-side: the PT translational errors. Right-side: the PT rotational errors. b) The ST performance. The blue graph represents the baseline. The end-effector positions are shown in the first three columns, whereas the 3D trajectories are presented in the last one.

$p_{NASA} = 0.1$). Even though the limited number of users did not allow to obtain statistically significant results, some preliminary conclusions can be derived using the collected results (the effect sizes $d_{SUS} = 0.51$ and $d_{NASA} = 0.63$ are considered to be a large effect). Concerning the usability scores (S), both $\mathrm{I_{VR}}$ (S=80) and $\mathrm{I_{EVRR}}$ (S=71) seemed to be valuable solutions, whereas $\mathrm{I_{EVR}}$ provided unsatisfactory outcomes (S=58). Results that appear to be supported by the workload scores (W) ($\mathrm{I_{VR}}$ (W=34), $\mathrm{I_{EVRR}}$ (W=39), $\mathrm{I_{EVR}}$ (W=60)), indicating that the pure point cloud seems to be inefficient to teleoperate a manipulator. On the contrary, a virtual representation of the robot seems to greatly foster the interface usability. Concerning PT, it is clear that translational errors are minimal for $\mathrm{I_{VR}}$, followed

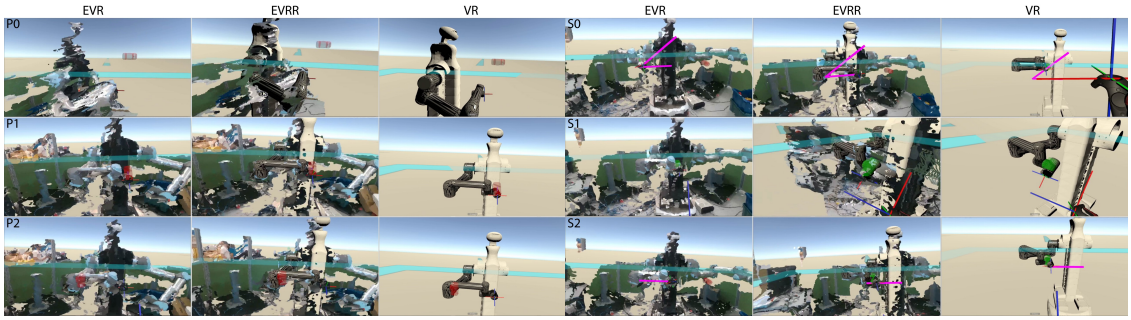


Figure 2.36: The pose tasks (left) and speed tasks (right). Task types are grouped in rows and the interfaces are grouped in columns. In the pose tasks, the Ghost is represented by the red virtual end-effector. In the speed tasks, the users have to move the robot end-effector along the trajectory when the Ghost turns green, matching its velocity.

by I_{EVRR} and I_{EVR} (Fig. 2.35a). In contrast, the rotational errors seem to be quite high, independently of the employed interface. Finally, although the ST outcomes show similar trends (columns 1-3 in Fig. 2.35b), the I_{EVRR} trajectories appear to match the baseline more closely than others (column 4 in Figure 2.35b).

These preliminary outcomes indicate that a pure point cloud interface seems to be less effective than interfaces that provide the visualization of the robot CAD model. To truly assess the interfaces effectivenss, a more rigorous study is presented in the next section. In addition to the evaluated parameters (translational and rotational errors for PT, translational errors for ST, usability and workload), the effects of the point cloud visualization on the teleoperation itself have been also considered. Moreover, more compelling ST tasks have been evaluated, considering also rotational and roto-translational trajectories.

2.6.5 The User Study

Eighteen new users have been involved in the teleoperation assessment task. In addition to PT (see Sec. 2.6.4), the users were asked to perform more compelling ST tasks and to fill a more rigorous questionnaire. Specifically, in addition to S0 (pure translational trajectory), two new speed task have been evaluated: trajectory with only rotation (S1) and trajectory with roto-translation (S2). For S2, the trajecotry has been highlighted using a virtual purple line similar to the one used in S0 whereas the S1 trajectory is not displayed because it consists of a pure rotational task (the Ghost only rotates along its horizontal axis with a pre-computed speed). As for the preliminary study, both PT and ST have been pre-recorded using the real robot, ensuring a common base line. Figure 2.36 shows both PT and ST visualized with the different interfaces.

Both objective and subjective data have been collected and analyzed. The objective data are (i) end-effector position accuracy (PT) and (ii) similarity among trajectories (ST). The subjective ones have been assessed through a questionnaire divided in: (Q0) users' age, sex, and familiarity with robotics and virtual reality (5-point likert scale), (Q1) the Simulator Sickness Questionnaire (SSQ) [217], (Q2) the System Usability Scale (SUS) Questionnaire [48], (Q3) the NASA Questionnaire [169], (Q4-Q5-Q6) the "Attention", "Spatial Situation", and "Presence" sections of the MEC Spatial Presence Questionnaire (MEC-SPQ) [465] to evaluate the virtual environment from a user perspective, (Q7) a custom ranking section (7-point likert scale) to evaluate the complexity of the three operations: translation, rotation, and roto-translation, (Q8) the Single Ease Questionnaire (SEQ)[23] to evaluate the overall complexity of the tasks and (Q9) a free form for comments.

Before doing the experiment, each user filled Q0 and he/she was given time to familiarize with the systems, trying to control the robot arm. Then the user filled section Q1 before trying a particular interface. After that, he/she ran both PT and ST for a specific interface, filling sections Q2-Q3-Q4-Q5-Q6-Q7 once completed all the tasks. The procedure is repeated with the remaining interfaces, changing every time the interfaces' order. Finally, the user fills Q8-Q9.

2.6.6 Results

In this section, the collected results are presented and discussed.

The Objective Results

	Interfaces			**Wilcoxon (p), Effect Size (r)**	
	VR	**EVR**	**EVRR**	**VR-EVR**	**EVRR-EVR**
P0T	0.007	0.0327	0.01	p=0.0, r=0.877	p=0.0, r=0.846
P1T	0.009	0.033	0.015	p=0.0, r=0.877	p=0.002, r=0.723
P2T	0.009	0.048	0.062	p=0.0, r=0.877	
P0R	0.241	0.308	0.321		
P1R	0.365	0.442	0.319		
P2R	0.45	0.446	0.433		

Table 2.12: The PT results.

PT performance was assessed in terms of EE positioning accuracy, with respect to the baseline. The translational error was computed as the Euclidean distance between the user and the Ghost end-effector position. The rotational error was

[23] https://tinyurl.com/24p2x2dd

determined as the minimal angle between the user and the Ghost end-effector orientation. Table 2.12 presents the mean translational (T) and rotational (R) errors for PT. The data distribution has been evaluated with the Shapiro-Wilcoxon test, showing a non-uniform distribution for all tasks but for P2R (the rotational errors of task P2). Then, for the non-uniform data, the Friedman test ($p < 0.05$) has been used to check whether there were any statistically significant differences, followed by a Wilcoxon Signed Rank test with Bonferroni correction ($p < 0.017$) in case those differences were detected. For the uniform data, the one-way ANOVA test has been used instead of the Friedman test (the effect size has been computed as introduced in [448] for non uniform data and as explained in [367] for uniform data). The Friedman test showed significant outcomes in the translational data of PT ($p = 0.0$) and the Wilcoxon test reported statistically significant differences between I_{VR}-I_{EVR} and I_{EVRR}-I_{EVR}. As can be seen from Table 2.12, I_{EVR} was found to exhibit significantly larger translational error compared to the other two interfaces. For I_{VR} and I_{EVRR}, the errors were in the range of 1 cm, which is still insufficient for high-precision tasks like welding or surgery. No significant differences have been found among the interfaces in terms of rotational error, which was in all cases relatively high.

ST performance was evaluated by analyzing the similarity between the user trajectories and the robot baseline. This was determined by using the Dynamic Time Warping (DTW) algorithm [144], which computes an optimal match between two time sequences, with certain restrictions for preserving continuity. Translation and rotation were evaluated separately, where the chosen metrics were Euclidean distance and minimal angle offset, respectively. The trajectory tracking error was defined as the normalized alignment cost, that is, the sum of the distances between individual matched points, divided by the total number of points. The error was thus obtained in meters for translation and in radians for rotation, where a lower score indicates a higher degree of similarity between the user trajectory and the baseline. The obtained results are collected in Table 2.13. Regarding the data distribution, S0T, S0R and S2R show a non-uniform distribution and thus they have been subsequently analyzed using the Friedman and Wilcoxon with Bonferroni correction tests. On the other hand, since the data of S2T, S1T and S1R show an uniform distribution, they have been evaluated using the one-way ANOVA test and a two-tailed t-test ($p = 0.017$) with unequal variance. The Friedman test detected statistically significant differences for S0T, S0R and S2R ($p = 0.001$, $p = 0.0$ and $p = 0.001$, respectively) whereas the ANOVA test reported significant differences only for S2T ($p = 0.0$). For most speed task aspects, significant differences were found between the I_{VR} and I_{EVR}, showing that the pure virtual reality interface better supports accurate trajectory tracking. Similar findings were recognized between the I_{EVRR} and I_{EVR} interfaces for some of the speed tasks. Much like the pose tasks, the speed task results exhibit errors in the range of centimeters (translation) and several degrees (rotation).

	Interfaces			Wilcoxon (p), Effect Size (r) T-Test (p), Effect Size (r)	
	VR	**EVR**	**EVRR**	**VR-EVR**	**EVRR-EVR**
S0T	0.012	0.018	0.016	p=0.001, r=0.774	p=0.006, r=0.649
S0R	0.108	0.146	0.119	p=0.003, r=0.692	
S2T	0.012	0.028	0.013	p=0.0, r=0.596	p=0.0, r=0.587
S2R	0.128	0.213	0.112	p=0.005, r=0.662	p=0.0, r=0.846
S1T	0.006	0.016	0.001		
S1R	0.123	0.14	0.142		

Table 2.13: The ST results.

The Subjective Results

Table 2.14: The subjective questionnaire outcomes. The symbol "*" denotes that no statistically significant differences have been found.

	Interfaces			Wilcoxon (p), Effect Size (r)		
	VR	**EVR**	**EVRR**	**VR-EVR**	**EVRR-EVR**	**VR-EVRR**
Q2	83.333	58.472	79.861	p=0.0 r=0.830	p=0.0 r=0.80	
Q3	36.999	63.351	39.036	p=0.0 r=0.871	p=0.0 r=0.861	
Q4	48.235	42.764	47.764	p=0.014 r=0.584	p=0.005 r=0.662	
Q5	15.647	39.529	46.176	p=0.0 r=0.853	p=0.007 r=0.644	p=0.0 r=0.882
Q6	31.117	40.882	45.764		p=0.014 r=0.584	p=0.004 r=0.661
Q7-T	6.000	4.722	5.944	p=0.002 r=0.731	p=0.002 r=0.843	
Q7-R	5.166	3.111	4.555	p=0.001 r=0.773	p=0.001 r=0.754	
Q7-TR	3.777	2.000	3.277	p=0.002 r=0.714	p=0.004 r=0.686	
Q8	5.666	3.222	5.000	p=0.0 r=0.881	p=0.0 r=0.841	

According to the Q0 outcomes, all users were male, with an average age of 28.78 years. Regarding their previous experience with robotics and virtual reality, they showed on average a moderate knowledge of both fields (robotic = 2.57, virtual =

2.58). Table 2.14 shows the remaining questionnaire sections. Since Q1 results did not include any simulation sicknesses, they have been omitted from the table. All results of the different questionnaire's sections have been firstly evaluated with the Shapiro-Wilk test to verify the data distribution (normal or not). Since in each of the sections at least one of the interface reported non-uniform data, the Friedman test has been used to evaluate the differences among the interfaces in each section, followed by the Wilcoxon Signed Rank test with Bonferroni correction. Regarding Q2-Q3-Q4 results, both I_{VR} and I_{EVRR} performed significantly better than I_{EVR} (no difference has been found between I_{VR} and I_{EVRR}). Q5-Q6 results show a different trend. For both sections, I_{EVRR} was considered the best interface to increase the perception of the environment and the sense of virtual presence. On the contrary, I_{VR} obtained the lowest scores for both sections and no statistically significant differences were detected between I_{VR} and I_{EVR} for Q6. The Q7 outcomes indicate that the I_{VR} allowed the users to perform the operations in a more intuitive way than the other interfaces. Also in this case, I_{EVR} did not allow the users to easily control the robot, whereas they considered the effectiveness of I_{EVRR} in line with I_{VR}. Finally, the results of Q8 confirm the previous results. The users perceived I_{VR} as the most intuitive of all the interfaces that have been compared, followed by I_{EVRR} and I_{EVR}.

2.6.7 Discussion and Conclusions

Overall, the results definitely point out that the pure I_{EVR} interface is not adequate to finely control robotic manipulators in high accuracy tasks, whereas the visualization of the robot CAD model greatly improves the effectiveness of the interface.

One of the main issues of I_{EVR} was related to the resolution of the point cloud. That is, it seems that a pure point cloud is not enough detailed to clearly visualised small/medium size objects. Specifically, it generates artifacts close to the objects' edges (Fig. 2.37) that do not allow to clearly distinguish the object from the background. The 3D points placed very close to the edge are correctly colored using the color information of the background, but mistakenly positioned using the depth information of the edge points, generating artifacts all around the objects. This limitation greatly affects the visualization of the robot end-effector, making really hard the robot control. This outcome seems to go against some previous works [480, 479]. These works focused on pick-and-place tasks and problems related to the point cloud artifact have not been reported. Moreover, since the point cloud changes at every frame, it does not provide stable virtual assets. By visualizing it, the point cloud may seem unstable and always in movement. This continuous motion increases the cognitive workload required by the interface, lowering the users' attention. These drawbacks have also been detected in the users' comments. In any case, strategies should be pursued to improve the quality of the

point cloud, making it less noisy. As an example, the objects' of interest (e.g., the robot arm) should be detected in the real environment and substituted in real-time with the corresponding CAD model (including textures).

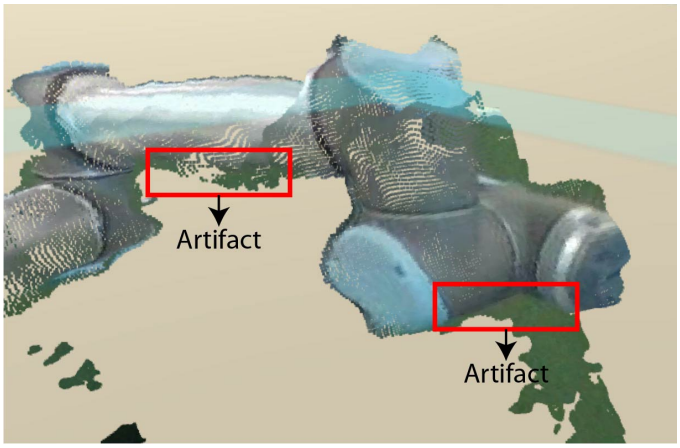


Figure 2.37: The point cloud artifacts generated close to the robot arm edges.

Concerning the "Spatial Situation" and "Presence" sections, since the users could only visualize the manipulator, unsurprisingly the I_{VR} interface has been considered not acceptable to foster the perception of the environment and the sense of immersion. However, it should be noticed that also in this case the I_{EVRR} has performed better than the I_{EVR} interface, probably giving a clearer visualization of the "main-actor" (the manipulator itself). Another concern is also the scaling factor between the user and the robot translation; a lower factor allows for positioning with higher precision, while a higher factor grants higher speed, requiring less human arm movements. This trade-off should be considered with respect to the specific task requirements.

At the current stage, I_{EVRR} is still inadequate for high-precision applications such as surgery or welding. However, it can be employed in creative tasks such as painting and choreography. It has also been demonstrated that the manipulator visualization could greatly improve interaction, obtaining performances similar to the I_{VR} interface. Future works will focus on improving the quality of the point cloud, trying to lower the negative effects of the artifacts. Moreover, the translation scaling factor will be added into the manipulator control interface, providing the users the ability to adjust the speed vs. accuracy trade-off during the task execution.

Chapter 3

AR VR in the Gaming Area

Part of the works done for this Ph.D dissertation and described in this chapter has also been published in [88, 86]. An additional work presented in Sec. 3.3 is currently under review at "Virtual Reality"[1]

Among the different VR and AR domains, the entertainment industry has always played a key-role in the advancement and improvements of these two technologies. A plethora of AR and VR applications can be found in the entertainment area [417, 20, 310, 266] and most importantly they can be used to develop videogames [486, 143, 133, 49, 385]. This fact is also confirmed by the remarkable income of this industry [464] and the aspiration of videogame players to be part of the game [26].

Since the nineties, VR HMDs have been used for gaming. One of the first well-know example is the Nintendo Virtual Boy, a 32-bit device capable of visualizing stereoscopic contents. Thanks to technological advancements, the VR headsets have begun to be employed in several different game-based domains. Even though the proposed applications are still based on the game logic, their main goal is not the entertainment itself, but the development of learning environments, thus involving educational experiences. On the current state of the art, several works can be found [32, 418, 449]. In [32], an immersive VR environment is exploited to evaluate what is the influence of perception over sport actions. Stone et al. [418] discussed the use of immersive headsets in the aviation industry, whereas VR immersive environments are employed in [449] for pain management. In the gaming area, further comparisons between VR and non-VR technologies were made to see how an immersive VR environment differs from a non-VR one and to determine the effect of immersive VR technologies on the game experience [331, 483]. In [331], different game experiences (with VR and non-VR technologies) are evaluated in a first person shooter game. Although the outcomes point out that the players preferred the

[1] https://www.springer.com/journal/10055/updates/18008620

non-VR interface in terms of usability, the VR one has been perceived as more compelling and attractive. Wilson et al. [483] analyzed the impact of immersive VR in violent video games. The collected results show how the users felt higher presence and body ownership with respect to the non-VR solution, feeling more violence and thus suggesting to change the game rating. Other studies related to VR can be found in [64, 66].

As well as VR, the AR technologies have been extensively used and researched in the gaming domain. The first acknowledged collaborative AR gaming system dates back to 1998, when Szalavári et al. presented a multi-player AR system for interacting with tabletop games [429]. A few years later, Thomas et al. [441] extended the Quake game to an AR scenario, allowing players to shoot down the virtual monsters in the real environment using a wearable AR device. The AR interfaces have been also researched to foster the game experience in outdoor environments by employing wearable [19] and handheld [72] devices. Even though the game design should be carefully planned to avoid undesired issues [40], new devices (such as the Microsoft HoloLens) have been successfully used in huge size environments providing exciting game experiences [369]. Referring to the indoor scenarios, Nojima et al. [315] presented an AR interface to augment sports by computer generated assets. A custom version of the dodge-ball game is presented whose players can fight each others by throwing a real ball. Players can visualize damages represented by virtual health-bars placed above the opponent players' head.

Concerning the AR and VR technologies in the gaming context, another interesting domain is represented by the "hybrid" games, that is, games that can be experienced using concurrently both AR and VR interfaces. Thomas et al. [442] improved their AR Quake game, giving support to AR/VR collaborative multiplayer, enabling players to impersonalize different virtual characters using a werable AR device and a desktop VR interface. Some years later, Cheok et al. [63] presented a modified version of the well-known Pacman game. Two players, using two wearable devices, were able to interact visualizing virtual assets superimposed in the real scenario. The VR player could provide support to the Pacman character represented by the AR player. A plethora of AR and VR devices is considered offering several different crossmedia experiences in [255]. Each interface provided users different functionalities, highlighting the peculiarity of each device. Clash Tank [358] presented a slightly different methodology. Although the users were placed in an immersive VR environment, they could still interact with the real scenario by visualizing it through a virtual monitor. Further examples can be found in [457, 117] which alternate the order of the AR/VR interfaces according to the game flow.

Considering the aforementioned state of the art, this chapter will discuss several works done for this Ph.D dissertation: (i) a preliminary study to evaluate the usability of the AR and VR interfaces for tabletop games, (ii) the impact of FoV on the usability of AR and VR interfaces in first person shooter games and (iii) a novel framework that eases the development of the hybrid games.

3.1 An Evaluation of VR/AR Interfaces Usability in Tabletop Games

The tabletop game has been chosen for several reasons: firstly, choosing otherwise would mean selecting a game that has been released for virtual reality and then porting it to augmented reality, or viceversa. However, porting a game to a specific environment (AR or VR) may result in a challenging task making it almost impossible to provide the same game experience in both environments. Secondly, several well-known tabletop games have been already ported into virtual reality environment, such as Scrabble, Monopoly or Risk. Among them, the chess game seems quite appropriate for multiple motives: (i) it is very well-known world-wide, (ii) it has already been employed as a test bed for research on AR game interfaces [363, 33] and (iii) it provides a strategic deepness essential to assess whether different interfaces could affect the game experience.

3.1.1 The System Architecture

In a chess game, only the start and the end positions of the piece that it is going to be moved by one player have to be exchanged, thus the proposed system is composed by two different reference systems that exchange data in real time on a socket connection. The transformations relative to the 3D assets (movement and rotations of the game piece) are applied locally in each system of reference.

The system architecture is composed by an Oculus Rift DK2 Kit (VR player) and the Microsoft HoloLens device (AR player). Both devices are connected on the same LAN. In the AR player environment, the opponent's game pieces (see Sec. 3.1.2) have been aligned using an image target placed at a known position with respect to the real chess board. Both AR and VR applications have been developed using Unity3D as game engine. In addition, the AR software includes the MixedRealityToolkit[2] to handle the interaction, the Vuforia library[3] for the target recognition and the LiteNetLib library[4] for the socket communication. On the contrary, the VR application includes the SteamVR Plugin[5] to access to the Oculus Rift DK2 hardware. The socket channel is used to exchange data related to the starting and ending positions of the game pieces. Hence, a move carried out in a specific environment is immediately replicated in the other scenario and viceversa.

[2] https://github.com/Microsoft/MixedRealityToolkit-Unity

[3] https://developer.vuforia.com/downloads/sdk

[4] https://github.com/RevenantX/LiteNetLib

[5] https://assetstore.unity.com/packages/templates/systems/steamvr-plugin-32647

3.1.2 The Game Play

In the VR environment both the game pieces and the chess board are virtual. On the contrary, the AR player can interact with his/her own real game pieces (the white ones) against the augmented representation of the VR game pieces (the black ones). The AR player starts the game changing the position of one of the real white pieces. At the same time, the virtual asset of the corresponding white piece is moved in the VR environment, keeping synchronized the two environments (see Sec. 3.1.3 for a detailed explanation regarding the movement of the white pieces). The same strategy is adopted when the VR player makes his/her own move.

3.1.3 The Interfaces

Two distinct interfaces (AR and VR) have been developed. Nonetheless, the interaction paradigm is identical for both interfaces and it is represented by the following work flow: (i) piece selection and (ii) piece movement.

The VR Interface

The VR player visualizes the game pieces by using the Oculus DK2 and he/she can interact with them using an XBOX 360 joystick. In order to select a specific game piece a combination of ray-cast and buttons is used. Specifically, when the ray-cast hits a virtual tile, the tile is highlighted in yellow color and a small virtual cube (called cursor) is rendered at the hit coordinates. If the tile contains a movable piece, the VR player can select it using the joystick "A button". Then, the available moves of the selected piece are highlighted on the chessboard. If one or more of the available moves intersect a tile occupied by an enemy piece, that particular tile is colored in red. The user can then select the final tile and the corresponding game piece is moved, ending the player's turn. Since the AR player can freely move around the real environment to visualize the chess-board from different points of view, the same feature has been added to the VR interface: the player can rotate the virtual chessboard around the global y and x axes analyzing the game field from other view points. Furthermore, since several chess games provide either a single top view or a 45° view, both view have been made available to the player through shortcuts.

The AR Interface

The AR interface is made up by three distinct layers: (i) gaze, (ii) gesture and (iii) sound layers. The gaze layer works as in the VR interface with just few adjustments. When the ray-cast hits a virtual hidden chessboard (not visible and aligned using the image target), a 3D cursor is rendered on the real chess board and the corresponding tile is virtually highlighted. In order to connect the real piece

and its virtual representation, a mechanism to synchronize the real move and the virtual one has been added to the system. When a tile containing a game piece is highlighted, the user can select it using the air tap gesture recognized by the HoloLens device, showing the available moves of the selected game piece. The tile can be selected using the air tap gesture and the available moves are shown on the real chess board. Then, he/she has to select the final tile by air tapping it and the player's turn ends when the real game piece is physically moved by the AR player. Every time an air tap gesture is performed, the position of the selected piece on the chess board is sent to the VR interface, keeping the environments synchronized.

To be sure the AR player physically moves his/her own game pieces, a pre-recorded voice informs the user to move the real piece. Moreover, a virtual green grid overlapped on the real chessboard can be activated or deactivated to clarify the visualization of the virtual pieces. Since animations can greatly improve the game experience, conveying emotions, motivations and intentions to the viewers [115], the "attack" and "death" animations of the game pieces have been added to both interfaces. Finally, a fixed user interface informs the user about the identity of the current player. Fig. 3.1 shows the AR and the VR interfaces.

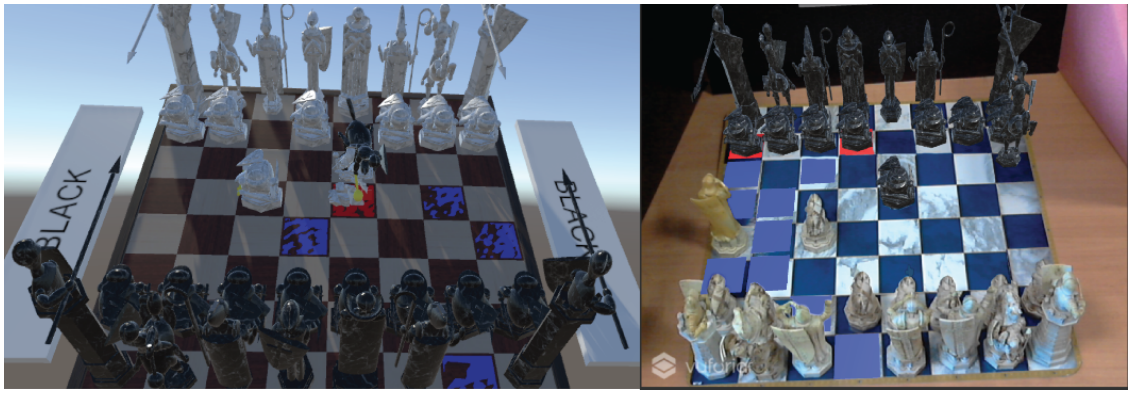


Figure 3.1: Left-side: the VR view. Right-side: the AR interface.

3.1.4 Tests and Results

In order to compare the usability of the proposed AR and VR interfaces, some tests have been carried out at the Politecnico di Torino. Twenty volunteers took part in the experiment, 12 men and 8 women, with ages between 21 and 34 years. The testers have been divided into 10 pairs, and each tester evaluated both interfaces. A questionnaire has been submitted to the users, using the System Usability Scale (SUS) [48] ranked with a five Likert scale. The test procedure was the following: given one pair, each user (user A and B) has been randomly assigned an interface (e.g., the AR one for user A and the VR one for user B). Then, both testers were

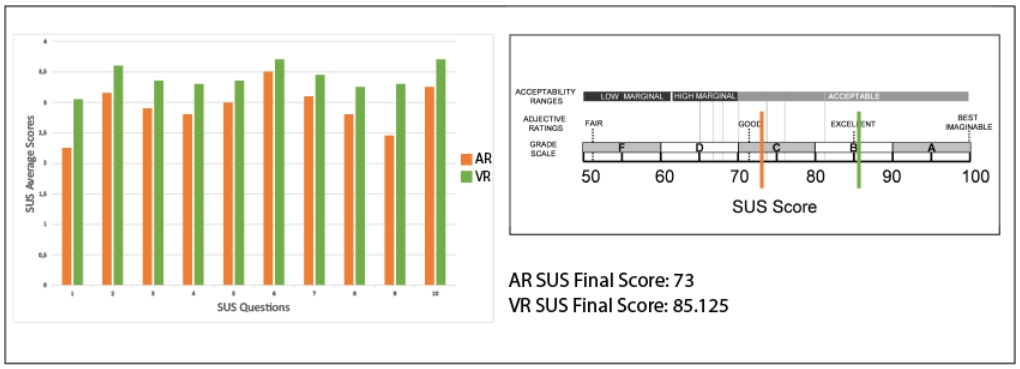


Figure 3.2: The SUS final scores.

given time to try the interfaces and the interaction paradigm. When the users felt ready, they could start the real experiment (the chess game), lasting at most 10 minutes. At the end of the game, each user had to complete the SUS questions related to the evaluated interface (either the AR or VR one). Then, the users swapped the interfaces, playing another training session. The last game session started, lasting again at most 10 minutes. Finally, each user had to fill the SUS questions related to the assessed interface.

Overall, both interfaces have been considered adequate to play the chess game, thus it is reasonable to consider the interface deployment through the proposed framework successful (Fig. 3.2). The SUS scores of the AR and VR interfaces are equal to 73 and 85.125, respectively, and the post-hoc Wilcoxon Signed Rank test showed a statistically significant difference between the interfaces ($p = 0.007$ with effect size $d = 0.6$, which is considered to be a medium effect size). The d value has been computed following [448]. It is possible to infer that the AR interface was affected by at least three different problems: (i) hardware related problem, (ii) interaction problem and (iii) visualization problem. The first issue is related to the very limited HoloLens FoV (around 35°) that prevented the users from clearly visualizing the moves of the VR opponent, loosing the game flow. The underlying reasons of the second issue are related to a weak link between the virtual environment and the real one. In fact, the AR input interaction forced the user to interact firstly with the virtual world and then with the real one. It is reasonable to presume that this "double" form of interaction required a substantial cognitive workload forcing the user to focus only on the interaction paradigm and not on the game itself. Moreover, some users had trouble doing the HoloLens tap gesture. The last issue regards the difficulty in perceiving the game field depth. When a virtual game piece was occluded by a real one (or viceversa), the AR users were not able to realize which piece was in front of the other and they had to change their position to visualize the game board from a lateral view.

3.1.5 Conclusions

A innovative AR/VR multi-player game system that allows players to experience the same (or similar) game experience has been presented and evaluated. Although the collected results do not allow to present clear and definitive conclusions, both interfaces were deemed suitable to interact with the same digital contents. Even though the interfaces belonged to different environments, the same functionalities were provided. If for a specific environment it is possible to obtain a particular interaction thanks to the software, the same interaction can be obtained in the other environment exploiting the hardware. However, a greater number of users should be involved to confirm the discussed hypothesis. Future works will analyze the effects of the FoV on different types of video games, such as the first person shooter games.

3.2 The FoV Impact on Hybrid First Person Shooter Games

Few works have analyzed the impact of FoV on AR applications. In [456], three distinct FoVs (small, medium and large) have been compared in target following tasks. The outcomes show that small FoVs are inadequate for tracking people whereas no significant differences have been found between medium and large sizes. On the contrary, Ren et al. [366] found out that a large 108x82° FoV allows users to fulfill tasks quicker than by using a small FoV constrained to 45x30°. Two different typologies of AR labelling techniques (*in-view* and *in-situ*) have been compared using a custom display with dynamic FoV, in a search target scenario in[225]. The main results indicate that as the FoV reaches 100°, the in-situ labelling discovery rate increases, whereas the in-view one shows an opposite trend. Nonetheless, the performance of both strategies converges at 130°. Finally, an evaluation of the cognitive workload using three distinct devices with different FoVs in a button-pressing precedural task is presented in [27]. The outcomes prove that the projected AR device required substantial less cognitive load compared to the other devices. Hence, the authors suggest using a great number of visual aids for improving the performance of narrow FoV devices.

Moving from the above analysis and from the main outcomes of Sec. 3.1, there is a lack of works regarding the impact of FoV on the usability of hybrid multiplayer first person shooter games. In the next sections, the system architecture and the proposed strategy are presented and discussed.

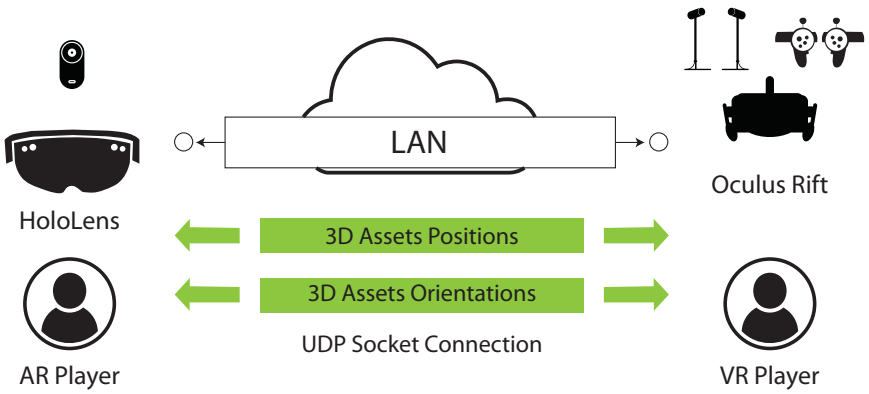


Figure 3.3: Left-side: the AR player with the HoloLens device. Right-side: the VR player with the Oculus device.

3.2.1 The System Architecture

The device of the VR player (VRP) is represented by an Oculus Rift (and related Touch controllers) connected to a Personal Computer (PC), whereas the AR player (ARP) can interact using a Microsoft HoloLens[6] (and related Clicker[7]) device. Both devices should be connected on the same LAN using a UDP socket connection.

The Unity3D game engine has been used to develop the hybrid environment. The VRP PC acts as a server (specifically as a host) whereas the ARP device acts as a client. In order to manage the network connection and to access the data from the two employed devices, the following libraries have been used: (i) the MixedRealityToolkit-Unity[8] and SteamVR Plugin[9] to access the HoloLens and Oculus hardware data, respectively and (ii) the Unet Unity API[10] (High Level API) to manage the client-server architecture. Figure 3.3 shows the system architecture.

3.2.2 The Game Level Design

To single out the influence of the FoV, the same game experience should be conveyed, overcoming the intrinsic differences of the employed devices. To achieve the proposed goal, the level design, that is, the set of game level, user interface and

[6] https://www.microsoft.com/it-it/hololens

[7] https://docs.microsoft.com/en-us/windows/mixed-reality/hardware-accessories

[8] https://github.com/microsoft/MixedRealityToolkit-Unity/releases

[9] https://assetstore.unity.com/packages/templates/systems/steamvr-plugin-32647

[10] https://bitbucket.org/Unity-Technologies/networking/src/2018.3/

alter ego, should be carefully planned to make the VRP and ARP game experiences as similar as possible.

The Proposed Alter Ego

Figure 3.4: The virtual drone controlled by the players.

A virtual drone has been used as alter ego for both VRP and ARP for two main reasons: firstly, the control of human characters would have required an ad-hoc input system, emphasising the devices differences. Secondly, a humanoid avatar could be arduous to visualize using the HoloLens device (see Sec. 2.5). Thus a simpler and more visible alter ego has been used. The virtual drone is shown in Fig. 3.4. It can fly in all directions, firing from two laser side guns.

To ensure that both players can control the alter ego in a similar way, the player's inputs have been carefully planned. Specifically, concerning the ARP, the HoloLens global position and the local rotation have been directly mapped to the virtual drone's global position and local rotation. Thus, the drone can reach at most an altitude equal to the ARP's physical height. The laser fire action is handled using the direction of sight as a gun-sight and the HoloLens Clicker button as a fire button. The HoloLens gesture recognition mechanism has not been employed as hands may occlude the fire direction and the air tap gesture can strain arm muscles.

The VRP's interaction paradigm has been designed in a similar way. The left Oculus Touch thumbstick and the rotation of the headset have been mapped to the drone global translation and local rotation, respectively. Specifically, the head rotation provides the motion direction, whereas the vertical axis of the thumbstick provides the translation magnitude. On the other hand, the thumbstick horizontal axis provides both the magnitude and the direction of translation. A maximum

drone flight altitude has been added as a constraint to provide movements comparable to those of the ARP. Finally, the right Oculus controller trigger button is used to shoot the laser bullets. Figure 3.5 illustrates the ARP and VRP input mapping.

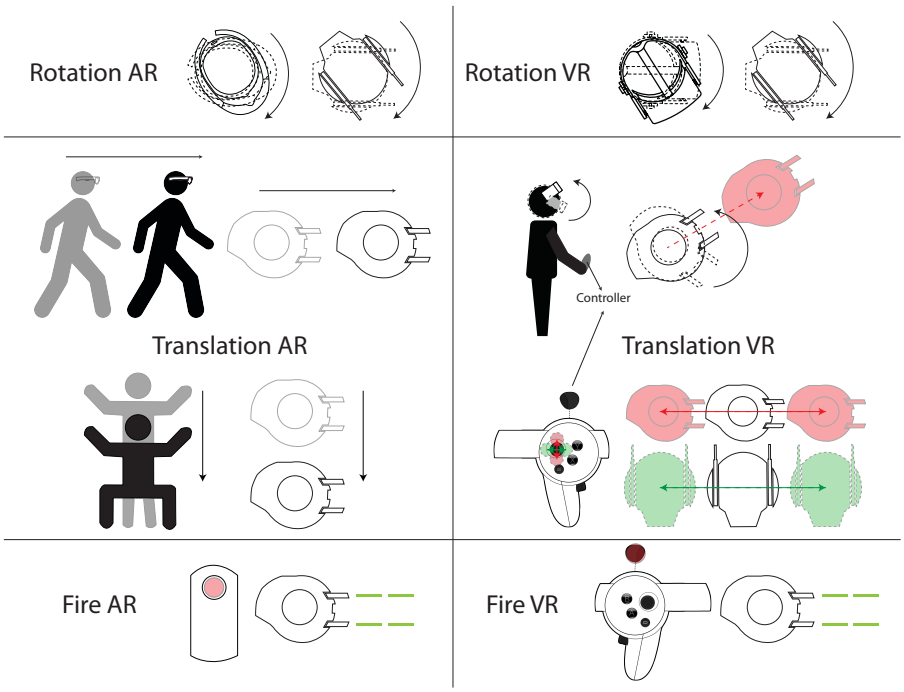


Figure 3.5: The similar input mapping between AR (left) and VR (right)

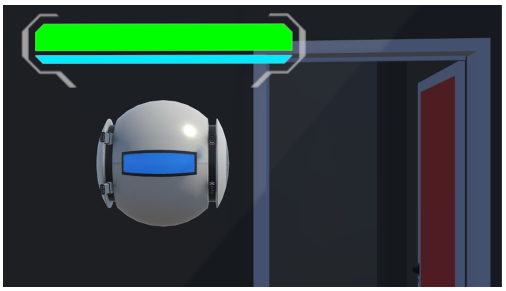
.

The Proposed User Interface

The user interface has been kept as minimal as possible to provide a wide field of vision. It is essentially composed by three elements that represent the life of the players, the action of being shot and the action of firing.

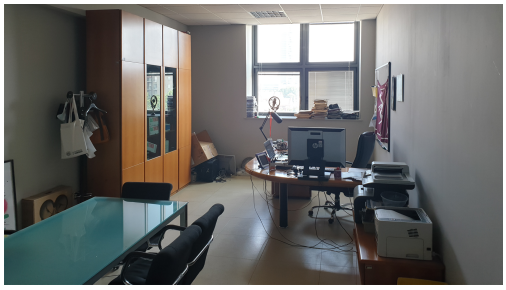
The drone life is represented by a virtual health-bar, placed at a pre-defined distance from the top-part of the drone. To make players aware of being shot, the virtual camera is occluded for few seconds with a semi-transparent red panel, highlighting the shoot action. Finally, when the laser bullet is shot, a sound of gunfire is played improving the realism and the sensation of game immersion. Figure 3.6 shows the two different user interfaces.

(a) The AR view.



(b) The VR view.

Figure 3.6: The virtual UIs.



(a) The real office.

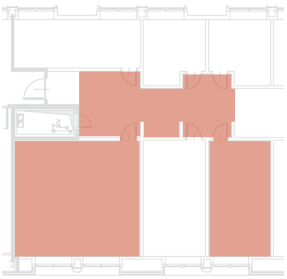
(b) The virtual representation of the real office.

Figure 3.7: The real and virtual environments.

The Hybrid Environment

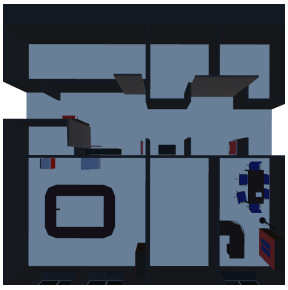
The game environment consists of two separated rooms connected by a hallway at Politecnico di Torino university. Using a detailed map (with dimensions, obstacles, etc.), a designer has modeled the related virtual environment, matching 1:1 the obstacles positions/orientations (Fig. 3.7).

Since the networking layer is represented by a client-server architecture, the game environment has been deployed on the server (the VRP system, acting as a host) and the ARP can access it by using the UDP socket connection. However, to be consistent with the different interfaces, only the VRP can visualize the virtual environment, whereas in the ARP's scenario the virtual meshes have been kept invisible. Only some virtual obstacles and traps have been rendered in the ARP's environment, improving the game realism. Furthermore, a pre-scanned point cloud of the game environment has been used to improve the HoloLens spatial mapping capabilities [369]. The related virtual mesh has been only deployed in the ARP's client application. Figure 3.8 shows the game environment and the related point cloud.

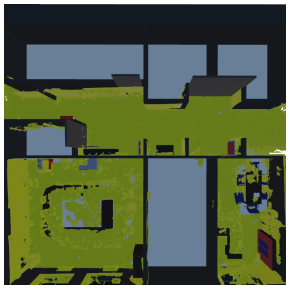


(a) The detailed map of the real office. The red areas denote the game field.

(b) The office point cloud.



(c) The virtual game map.



(d) The point cloud superimposed over the virtual office.

Figure 3.8: The game map.

3.2.3 The ARP Environment Improvements

Since ARP is acting in the real environment, there may be issues related (i) to the alignment of the reference system, (ii) to the restriction of the game map and (iii) to the occlusions. The former has been managed defining a fixed ARP starting position, that is, before starting the game, the HoloLens device has to be placed on the real floor with a known position and orientation (Fig. 3.9 top-left) to complete a calibration procedure.

The second one refers to the real physical boundaries of the game environment. To prevent the ARP from entering in zones not included in the game environment, some virtual models have been added to the game map (see Fig. 3.9 top-right). They have been placed in front of doors or corners in which the ARP should not pass. These specific assets have been chosen because they remind the concepts of "forbidden", "work in progress" or "prohibition".

Finally, the point cloud color material of the game environment (see Sec. 3.2.2) has been set to pure black to properly handle the occlusions between the real and virtual model. Hence, the ARP is able to visualize the real environment and the occlusions between the game map and the virtual models (Fig. 3.9 bottom-left and bottom-right).

(a) The HoloLens starting position.

(b) A virtual barrier to avoid ARP entering in non-game zones

(c) The point cloud black material.

(d) The occlusions between real and virtual objects.

Figure 3.9: The ARP environment improvements.

3.2.4 The Game Modality

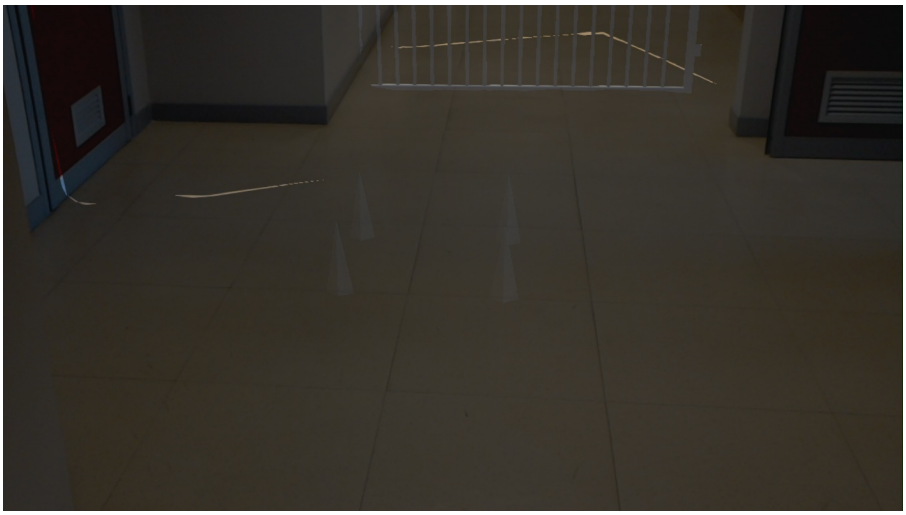


Figure 3.10: An example of trap added to make the game more compelling.

The players compete one against the other, trying to eliminate the opposing

drone. Some virtual traps have been added to the game environment, making the game more attractive and compelling. The traps are placed in some specific zones and they can be activated when a player enters in their operative zone (Fig. 3.10).

3.2.5 Tests and Results

The preliminary tests involved ten people divided in five sessions; two users evaluated the system for each session. A test session is split in two different sub-sessions. During the first one, one of the users plays with the VR device and the other one plays with the AR device. Then, after 10 minutes, the users have to fill the SUS usability questionnaire [48]. During the second sub-session, the users exchange the devices and play a new game session for 10 minutes, filling the SUS questionnaire once the game match ends. Once completed both sessions, the users have to fill a custom section indicating pros and cons of the VR and AR interfaces. Users were Ph.D. or M.Sc. students of the computer science department and they had some previous experiences with the AR and VR technologies. Hence, the training phase was kept quite short and was basically aimed to explain the shooting mechanism. The users involved in test were 9 males and 1 female, with age ranging from 21 to 36 years.

Tables 3.1 and 3.2 show the AR and VR outcomes, respectively. Mean value μ_{AR} and variance σ^2_{AR} for the AR application are 27.3 (68.5 normalized in hundredths) and 67.3, respectively. Whereas μ_{VR} and variance σ^2_{VR} for the VR application are 30.5 (76.25 normalized in hundredths) and 23.1, respectively. Under the null hypothesis H_0 that $\mu_{AR} = \mu_{VR}$, and the alternative hypothesis that the two means are different, the Wilcoxon Signed Rank Test has been performed by obtaining a probability $p = 0.386$ higher than 5%, therefore the null hypothesis cannot be rejected and the difference between the two mean values of usability is not statistically significant (the effect size $d = 0.27$[11] can be considered to be a small effect). It cannot be claimed that an application is more usable than the other one, even if a slight preference for the VR solution can be noticed. Despite these limited outcomes, the majority of the users indicated the FoV as the main limit of the AR interface (as in Sec. 3.1): it was very hard identifying the enemy position, specifically when the two opponents were close. On the other hand, the possibility of moving in the real environment is the most appreciated aspect of the ARP interface. Referring to VR, the users appreciated the level of realism of the environment but they found difficulties in using the gaze pointing mechanism. Only one user slightly experienced motion sickness using the VR application.

Finally, only two AR users exceeded the game area, indicating that the virtual barriers design was effective to limit the ARPs' movements.

[11] The effect size has been computed following the approach detailed in [448]

User	Q_1	Q_2	Q_3	Q_4	Q_5	Q_6	Q_7	Q_8	Q_9	Q_10
1	3	4	3	4	3	4	4	4	4	4
2	1	3	2	4	2	2	3	4	2	4
3	2	3	3	1	3	4	2	3	1	1
4	1	4	3	3	3	4	3	2	3	4
5	1	4	3	4	2	2	3	3	4	4
6	2	4	3	4	3	4	3	4	3	4
7	2	2	1	2	2	3	3	2	3	4
8	2	2	4	1	3	3	4	3	4	4
9	0	1	0	1	3	2	0	0	0	1
10	1	4	4	3	1	3	4	4	2	4

Table 3.1: Results obtained by testing the augmented application.

User	Q_1	Q_2	Q_3	Q_4	Q_5	Q_6	Q_7	Q_8	Q_9	Q_10
1	1	3	3	4	2	2	3	3	3	4
2	3	4	3	4	3	2	3	3	3	4
3	3	3	3	2	3	4	2	4	3	3
4	3	4	3	3	3	4	3	4	3	4
5	3	4	3	3	2	2	1	2	2	4
6	2	3	3	3	3	3	2	3	2	3
7	2	2	3	2	2	1	2	2	3	4
8	4	4	4	4	4	4	4	4	4	4
9	3	4	3	4	3	4	3	4	2	3
10	1	4	4	3	3	2	3	4	4	4

Table 3.2: Results obtained by testing the virtual application.

3.2.6 Conclusions

The usability of AR and VR interfaces has been assessed in a first person shooter game. Although it has not been possible to statistically demonstrate the effects of different FoVs on the game usability, several users reported that the FoV was the major cause of issues, lowering the AR game experience (results similar to Sec. 3.1). Nonetheless, they have appreciated the possibility of moving freely in the real environment. It is worth mentioning one possible limitation of the proposed study. Although great effort has been done to convey the same game experience by providing almost the same interaction paradigm, the devices intrinsically differ from each other and it is possible that other factors have affected the game experience. Further experiments will be carried out by using devices that provide more similar characteristics than the ones used in the proposed experiment. For instance, a HTC Vive Pro equipped with the external video cameras could be used to convey both VR and AR experiences.

3.3 The VR/AR Framework

The previous chapter presented a consistent state of the art related to the use of AR and VR interfaces for hybrid environments. However, the creation of such environments requires great effort and skills and there is a lack of study or projects that provide frameworks to ease the development of the hybrid environments. When developing hybrid scenarios, some design and implementation issues may arise due to the intrinsic differences between these two technologies. Since the AR players should be able to play in an augmented version of the real world, the virtual assets (commonly referred to as *entities*) should be merged seamlessly with the real ones. Furthermore, the AR physical movements should safely take place in medium/large size areas, whereas the VR movements are usually constrained in relatively small and obstacle-free zones. As it has been explained in this chapter, to let both users play in the same hybrid environment, it is necessary to create a virtual replica of the AR environment or at least one that is designed coherently with the obstacles and the objects positioned in the play area, blending elegantly the virtual assets with the real objects. Whereas several AR users physically placed in the same location would be able to fully see each other, the VR players, limited by the current technologies, would visualize just an approximation of the real users (in terms of appearance and motion). Especially for video games, being able to effectively and equally interact with the other participants is not only important from a usability point of view, but it also guarantees a fair game experience.

To develop a hybrid multi-user application, developers have to usually employ SDKs (often not open-source and protected by licenses) specific for a particular device. Furthermore, Broll et al. [47] identified a set of common issues that should be carefully taken into consideration in order to develop a fully immersive experience: (i) keeping shared worlds consistent, (ii) the network protocol must scale to the (large) number of users, (iii) consideration of reliability issues versus interactivity, (iv) support of cooperation rather than coexistence, (v) heterogeneous network connections, and (vi) composition of large-scaled subdivided worlds.

NPSNET [271] and VRML 2.0 [54] represent two early works in the sharing of virtual contents over the network. However, they are not suitable for modern computer applications and devices. DIVERSE is another example of modular VR framework [216]. Although its aim was to ease the development of device-independent virtual scenarios, it was not designed with AR in mind. VHD++ [348], MORGAN [319] and Instantreality [29] are modular and extensible frameworks for AR and VR. Although they are indeed quite powerful frameworks, the underlying technology is quite complex and they do not provide integrated and easy to use tools for scene creation and management. Other examples of frameworks targeting only VR are represented by inVRs [13], CalVR [389] and CocoVerse [391]. The first one

consists of a C++ framework for networked VR applications offering a modular design, the second one is an open-source VR middleware based on OpenSceneGraph[12] whereas CocoVerse is a multi-user immersive environment where users can collaborate to create virtual assets using a set of predefined tools. Despite their promising capabilities, CalVR, inVRs and CocoVerse have been designed for VR and they are not suitable for hybrid environments. Finally, ARTiFICe [298] is one of the most recent frameworks with multiplatform support and integrated authoring tools for scenes creation. To the best of the authors' knowledge, ARTiFICe does not support modern devices such as the Microsoft HoloLens and it does not offer a uniform experience regardless of the hardware of choice.

Although the presented frameworks provide several functionalities and support extensibility, even the most recent ones (and other commercial frameworks such as the MRTK V2[13]) present at least two relevant issues: firstly, most of them are not designed to support a hybrid multi-player environment; secondly, even though some of them could theoretically provide such support, they are not designed to provide a comparable experience regardless of the employed hardware (VR or AR). Thus, in the following sections, the novel framework called *Harmonize* will be presented and discussed. Its most relevant novelties are the following: (i) developers can create hybrid environments providing a similar experience for both AR and VR players; ii) the framework is hardware-independent and (iii) its design is as extendable to novel hardware as possible. Moreover, Harmonize has been also evaluated collecting both objective and subjective parameters assessing its performance from a newtworking point of view as well as its usability from a user perspective.

3.3.1 The System Architecture

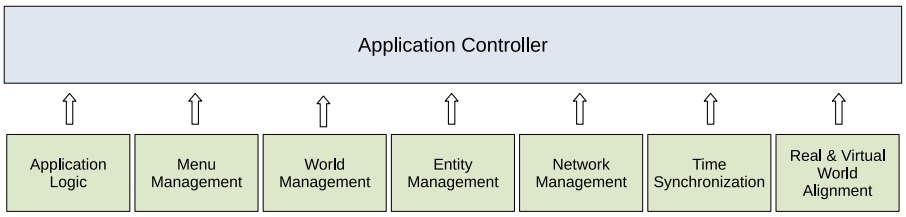


Figure 3.11: The framework architecture with application controller and modules. The Application Logic refers to the application mode, which is defined by a set of rules. The Network Management module handles connections and network messages. The Real & Virtual World Alignment module is needed to support AR systems.

[12] http://www.openscenegraph.org/

[13] https://github.com/microsoft/MixedRealityToolkit-Unity/releases/tag/v2.4.0

The system requirements are the following: firstly, the developers should be able to easily add both the rules of the intended application (e.g., the game rules) and new entity classes (entities are human-controlled or computer-controlled actors). Secondly, the framework should handle autonomously all the networked application features, such as connections, distributed computation and synchronization. The framework has been designed in a modular fashion (see Fig. 3.11). For every major functionality, such as networking and entity management, an independent module has been developed and all the modules are administrated and coordinated by a central unit, the application controller.

The User Input Management

Modern frameworks ensure the input forward-compatibility, that is, the applications developed with these SDKs will be compatible with upcoming devices. This can be obtained by virtualising raw inputs and providing actions that can be triggered by inputs. When a new device appears on the market and the toolkit is updated, the upgraded runtime will provide the proper input mapping, thus avoiding extensive code-rewriting to make the application compliant with the new device (the same approach is exploited by the MRTK, SteamVR and OpenXR frameworks). Thus, Harmonize provides the same forward-compatibility of the other frameworks and it automatically detects the user device type, allowing users to choose the preferred interaction paradigm among those supported by the selected hardware.

The Shared World Structure

The hybrid environment should be topologically similar to the real play space. There are at least two different strategies to virtualize the real environment: (i) on the fly reconstruction or (ii) 3D modelling, beforehand, the real location. The first approach requires techniques that are highly computationally expensive (e.g., KinectFusion [198]) for the current AR devices; moreover, the reconstruction level of details may not be sufficient to provide an immersive VR experience. Hence, the actual version of Harmonize supports only traditionally generated environments.

The VR Locomotion Method

There exist several locomotion methods (see Sec. 1.2.1 for a detail explanation of the different locomotion methods) designed to allow users to walk in the virtual world, some of which are better than others at preventing motion sickness [288]. Among all possible locomotion methods, Harmonize supports the arm swing technique, allowing users to virtually walk by swinging their arms as if they were actually walking and supporting tethered VR devices. Hence, the VR players can virtually move in large environments without experiencing cybersickness

symptoms and matching the walking capabilities of the AR players. The arm swing strategy has been added to Harmonize integrating an existing library named ArmSwinger[14].

World State Synchronisation and Network Model

Multi-user applications usually rely on distributed models: the users are remotely connected and the computation is not centralized but distributed among multiple machines. To comply with this specific architecture, two main mechanisms should be supported: (i) host network communication and (ii) application state synchronization. The proposed framework relies on a client-server architecture, easing the synchronization process: the clients send data and inputs to the server which in turn elaborates them sending back the new state of the world to each client. This architecture provides several advantages: (i) it scales easily as the number of clients increases, (ii) less powerful AR devices do not have to deal with heavy computational tasks, which are instead carried out by the central server and (iii) the development of a client-server architecture for multi-user applications requires less effort with respect to a *peer-to-peer* architecture.

3.3.2 Implementation

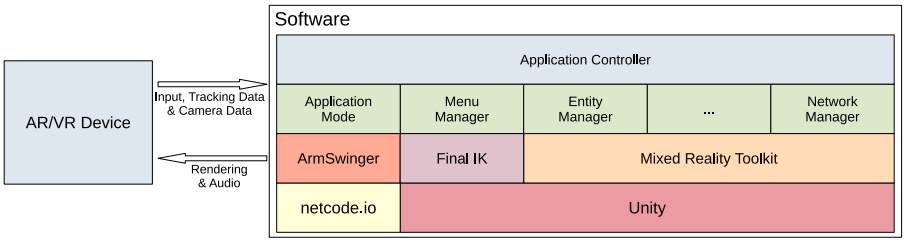


Figure 3.12: The arrows indicate the data-flow between the software layer and hardware devices

Harmonize has been developed using the Unity3D game engine. Unity3D is flexibile and it can be easily integrated with third-party plugins. Figure 3.12 shows the software architecture, highlighting the integrated third-party plugins (they will be discussed in Sec. “The Communication Protocol” and Sec. “Arm Swinger”).

The Communication Protocol

The Harmonize network layer relies on the UDP protocol. Since it is a connectionless protocol, UDP is fast and it is normally used to develop Massive Multiplayer

[14] https://github.com/ElectricNightOwl/ArmSwinger

Online Games (MMOGs). It greatly helps reducing the packet lag and jitter that can negatively affect the game experience, especially for VR and AR applications whose visual imperfections can be easily recognized by the users [31]. However, the UDP speed comes at a cost: it does not provide some basic features, such as verifying whether a remote host is still reachable and can communicate with the local host. In order to add custom networking funcionalities, Harmonize adopts the netcode.io[15] third-party library, a connection-oriented protocol built on top of UDP and designed for high-performance and low-latency videogames.

World Synchronisation Issues & Solutions

In order to reduce the latency and to foster the perceived fluidity of the virtual environment, Harmonize employs three different techniques that are usually implemented in current MMOGs. The first one is called *client-side prediction* and it allows to compute the user's input at the client side, instead of sending it to the server and waiting for its response. Although this technique helps reducing the lag (the users start seeing inconsistencies if the lag is greater than $50ms$ [353]), the client response may differ from the server one. Hence, the *server reconciliation* technique is usually employed to reconciliate the client with the server, that is, the client adjusts its current state making it coherent with the one determined by the server. To further improve the overall fluidity, Harmonize adopts a third technique called *entity interpolation*. The clients can use the past entity states interpolating their last position and orientation with the last received state. Hence, the server has time to bufferize the client inputs and to process them all at once at low frequency (e.g. 10-20 times per second), using less CPU resources.

The Network Manager

The network manager is a module built on top of netcode.io (see Sec. 3.3.2). It is used every time to create a client-server connection or to exchange network messages. A dedicated thread manages both events and messages. The Network Manager provides dedicated methods to start or end a connection and to send messages. When required, the Network Manager can deal with message fragmentation. Moreover, since UDP is unreliable, the Network Manager extends it with a reliability layer for all those cases where it is strictly necessary.

Virtual and Real World Alignment

For the VR user, the alignment of the player with the virtual world is done by the device itself or by using external sensors. On the contrary, the AR player

[15] https://github.com/networkprotocol/netcode.io

alignement relies on the use of the so called *anchors* [244], which store color or shape features of a given real location. Then, this information is later recognized at run-time when the AR device recognizes these specific locations, computing a transformation used to correctly align the detected anchor with its virtual location in the game scene. In case two ore more anchors are detected at the same time, the framework considers the closest one to compute the alignment transformation.

Arm Swinger

The arm swinger technique allows users to virtually walk by swinging their arms. The movement direction can be controlled by the hands' rotation whereas the magnitude speed can be changed by varying the swinging frequency. To further improve the locomotion strategy, if a collision between the player and a virtual obstacle is foreseen, the system automatically slows down the player, preventing the collision.

Interaction Methods

At the current stage, Harmonize targets wearable VR and AR devices. VR HMDs usually provide controllers that track the real hand's position and orientation whereas the wearable AR devices (such as the Microsoft HoloLens) provide gesture recognition systems to detect the user input. Hence, both AR and VR players can interact with the digital contents using a similar approach. The players have to firstly gaze at the desired object. Then, the VR user can virtually move his/her hands inside the virtual asset, whereas the AR player has to perform the air tap gesture to interact with the virtual models.

AR and VR Avatars

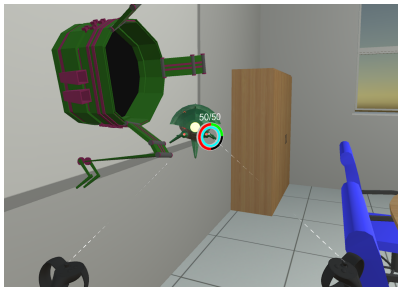
Harmonize integrates a third-party package called Final IK that allows both AR and VR players to see an animated virtual representation of other human characters. It is capable of animating the entire human skeleton, understanding the current player motion (i.e., whether the user is standing still, walking or running) by using the hands and head tracking data. Since the AR devices do not continuously track the hand positions, only the head tracking data are used to animate the AR virtual character in the VR environment. On the contrary, the AR real player will visualize a virtual representation of the VR user animated using both head and hands tracking data.

3.3.3 The Use Case

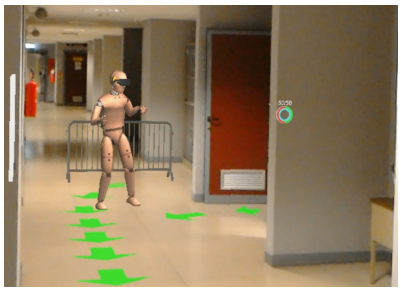
The proposed framework has been evaluated using the Microsoft HoloLens (AR player) and the Oculus Rift CV1 with Touch Controllers (VR player). The Oculus

devices was tethered to a PC workstation acting as the system server whereas the HoloLens device acted as an AR client. Both devices were connected on the same LAN. The test sessions was the same of the experiment introduced in Sec. 3.2. Figure 3.13a shows the virtual environment, whereas Fig. 3.13b - 3.13c illustrate the VR and AR views, respectively. The virtual anchors have been previously positioned in the real space. Since high contrast areas can be easily detected by the AR camera, the anchors have been placed in several wall corners of the play area.

(a) The virtual scenario.



(b) The VR view.



(c) The AR view.

Figure 3.13: The hybrid environment.

The players had to collaborate to destroy some virtual enemies positioned all around the play area in some fixed locations. Virtual medical kits and ammunition have been randomly added to make the game more competitive. The game session ended when the players destroyed all the enemies.

3.3.4 Tests and Results

Both objective and subjective paramters have been evaluated. Specifically the subjective ones include the usability (SUS [48]) and the game experience (Game Experience Questionnaire - GEQ [190]), whereas the objective data were the number of time the HoloLens lost the tracking, the time required to restore the tracking, the round-trip time and the number of lost packets.

Twenty users (4 females and 16 males), divided in pairs, were asked to assess

the Harmonize framework. Their ages were between 19 and 30 years and they all had previous experience with both AR and VR interfaces. Before starting the experiment, the users had been given time to try the AR and VR interfaces and the game play modality. After the first game session, the participants had to fill in the questionnaire section related to the evaluated interface. Then, they switched the devices for the second part of the test session repeating the game session and filling another time the remaining section of the questionnaire.

Concerning the SUS results (Fig. 3.14), the AR and VR interfaces obtained a similar positive score and they were both considered equally suitable to interact in the proposed environment (the Wilcoxon Signed Rank test showed a $p = 1$ with effect size $d = 0$).

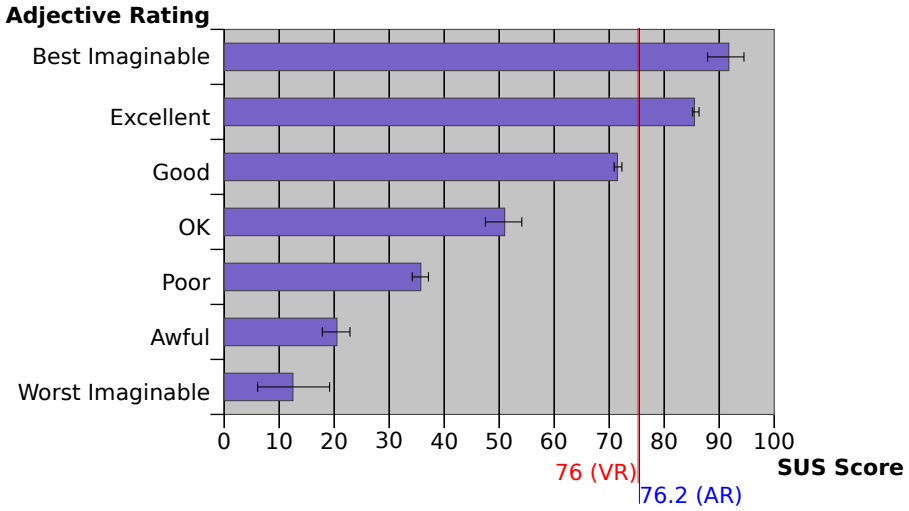


Figure 3.14: Correlation between the SUS score and adjective ratings [24]. The score of the proposed system is shown in the figure separately for VR and AR.

Following [190], the GEQ outcomes have been clusterized in the Game Experience Core Module, Social Presence Module and Post-game Module. As can be inferred from Table 3.3, the AR interface obtained lower scores than the VR one but it has been possible to detect statistically significant differences only for the Competence, Flow and Positive affect sections. Referring to the Social Presence Module (Table 3.4), both interfaces obtained relative low results and it has been possible to detect statistically significant differences only for the Behavioural Involvement category. Finally, the Post-game Module outcomes show that the VR interface generally obtained higher scores with respect to the AR one (Table 3.5). Specifically, it seems the users spent a much more positive experience with the VR interface than the AR one (the result is also confirmed by the post-hoc test that shows statistically significant differences). Referring to the statistically significant outcomes of Tables 3.3, 3.4 and 3.5, the post-hoc analysis shows small effect sizes

d for the related categories, thus suggesting that the VR and AR interfaces do not provide substantially different experiences.

	Competence	**Sensory**	**Flow**	**Tension**	**Challenge**	**Negative affect**	**Positive affect**
AVG AR	2.31	2.625	2.23	0.65	1.69	0.362	3.06
SD AR	0.17	0.45	0.83	0.05	0.8	0.27	0.2
AVG VR	2.76	2.667	3.07	0.383	1.93	0.337	3.3
SD VR	0.04	0.32	0.49	0.38	1.11	0.28	0.1
Wilcoxon	0.042	0.527	0.042	0.102	0.593	1	0.042
Effect Size	0.45	0.14	0.45	0.36	0.12	0	0.45

Table 3.3: The Game Experience Core Module outcomes.

	Empathy	**Negative Feelings**	**Behavioural Involvement**
AVG AR	1.475	0.84	1.47
SD AR	0.49	0.19	0.39
AVG VR	1.56	1.02	1.64
SD VR	0.64	0.45	0.37
Wilcoxon	0.248	0.223	0.027
Effect Size	0.25	0.27	0.49

Table 3.4: The Social Presence Module outcomes.

	Positive Exp.	**Negative Exp.**	**Tiredness**	**Returning to reality**
AVG AR	1.92	0.29	0.25	0.67
SD AR	0.41	0.27	0.07	0.56
AVG VR	2.33	0.28	0.5	1.2
SD VR	0.44	0.31	0.14	0.69
Wilcoxon	0.026	1	0.18	0.109
Effect Size	0.5	0	0.31	0.36

Table 3.5: The Post-game Module outcomes.

Some users additionally reported complains for the lack of an audio channel that would have allowed them to communicate during the game session. In addition, they found difficulties in perceiving the game environment using the AR device, confirming another time the well-known limitations of the wearable AR devices.

Regarding the objective data, the HoloLens lost the tracking approximately 1.35 times per session and it was capable of reacquiring it in $5s \pm 2.3s$. The collected

network round-trip time was on average about $200ms$ and the packet loss was less than 1%, data consistent with the current MMOGs (also the users did note report any inconsistencies during the game session).

3.3.5 Conclusions

The collected results show that Harmonize can be effectively employed to create hybrid environments that provide similar game experiences regardless of the employed device (AR or VR). Despite the VR interface has been generally preferred by the users, only few statistically significant outcomes have been detect, thus it is not possible to conclude that the users spent different experiences using the AR and VR devices. Considering the framework, since Harmonize is hardware and context independent, it can be easily adopted in several other domains. Future works will be focused on investigating novel interaction paradigms allowing users immersed in different realities of the Reality-Virtuality Continuum to collaborate together.

Chapter 4

Conclusion

This dissertation has explored and researched how the VR and AR technologies can be used in the Industry 4.0 domain, with particular interest for the HRC context. Furthermore, it has analyzed the usability of hybrid games considering both tabletop and first-person shooter games. Specifically, this dissertation started by exploring the different uses of AR in the Industry 4.0 domain, highlighting its effectiveness in the maintenance-assembly-repair, training, product control quality, HRC, and building monitoring areas (Sec. 2.1). Concerning the HRC domain, the discussion moved towards the different uses of AR with the robotic arms, proposing a classification of the most relevant works and thus defining three different macro-areas: Workspace, Control Feedback and Informative (Sec. 2.2). Thanks to this analysis, it has been possible to figure out that there is a lack of studies that have truly assessed the proposed systems from a user-centred perspective. Although the technical aspects are indeed fundamental when proposing new approaches or technologies, the virtual interfaces are strictly bounded to the human beings and, consequently, it becomes of primary importance involving the final users in the development process.

Moving from these considerations, this thesis proposed a static AR interface to display industrial robot faults (Sec. 2.3). Although visualizing the robot faults in the real environment could improve the fault management, the proposed interface was negatively affected by the narrow FoV of the selected wearable device. Hence, this dissertation described an innovative AR adaptive interface that overcame the main limitations of the static one (Sec. 2.4). By considering the movement of the user and the areas of the image plane occupied by the manipulator, the interface can place the virtual representation of the faults in areas always visible to the operator and not occluded by the manipulator. Then, the discussion moved towards the industrial training scenarios by proposing a hybrid interface that allows a remote trainer to provide instructions to a local trainee using both VR and AR technologies (Sec. 2.5). The main goal was to verify whether the visualization of a virtual avatar could provide better performance with respect to the traditional methods (e.g., the

abstract metaphors). Although the results are partially inconclusive, the preliminary outcomes suggest that the abstract metaphors allow the users to complete the tasks with less time and they are much more appreciated than the avatar-based interface. However, in no-audio condition, the avatar interface improved the sense of human-human collaboration. Finally, Chapter 2 concludes proposing an evaluation of an enhanced VR interface to remotely control a real robotic manipulator (Sec. 2.6). The main results show that a pure point cloud-based interface is not adequate to accurately control a robotic arm. On the contrary, the CAD visualization of the manipulator itself greatly improves the robot controlling.

Regarding the gaming context (Chapter 3), the discussion started by proposing a hybrid environment that allows different players (VR and AR) to compete against each other in a chess game (Sec. 3.1). The outcomes concerning the interfaces' usability indicate that the VR interface was far more appreciated by the users. Among the possible limitations of the AR interface, the very narrow FoV of the HoloLens device seemed to have negatively affected the game experience, preventing users to clearly visualize the virtual assets.

Considering the FoV limitations, this dissertation proposed then a user study to verify whether the FoV could affect the game experience in hybrid games that require wide physical movements from the players, such as the first-person shooter games (Sec. 3.2). Although it has not been possible to statistically ascertain the influence of the FoV, several users reported the narrow FoV as the main cause of limited game experience. The Chapter concludes introducing an innovative framework to ease the development of hybrid environments (Sec. 3.3). The framework has been evaluated in a first-person shooting game and the main outcomes indicate that the proposed system is able to convey the same game experience regardless of the employed interface.

Given an overview of all the projects presented so far, it is necessary to present and discuss the main limitations of the research done for this Ph.D. dissertation. The first limitation concerns the choice of the users for the user tests. Although the users involved in the user studies usually come from "technically sound" domains (e.g., engineering students, researchers in the computer science domain, etc.), they only partially represent the real population. Especially for the Industry 4.0 domain, the proposed systems have not been evaluated involving the real final users, that is, the technicians or operators who truly work in the factories or companies. A possible improvement would be to involve those operators to verify whether the collected results are still legitimate and consistent. Furthermore, it would be appropriate to involve a larger number of users than the one considered for this thesis, thus improving the statistical significance of the results. The second limitation is related to the absence of control tasks. For example, in Section 2.5, the users were trained to assemble a robotic hand in different AR settings, and these different settings were compared. In order to contextualize the results, it may be useful to test how well participants perform the assembly task without any training, or

with a trainer physically present (rather than available through AR). Such upper and lower bounds to task performance are important to understand the usefulness, successes and failures of any proposed system. The third limitation regards the analysis concerning the impact of the FoV on the usability of tabletop and first-person shooter games (Chapter 3). The two different evaluations (Sec. 3.1 and Sec. 3.2), have only compared two specific devices (HoloLens vs Oculus DK2 and HoloLens vs Oculus Rift, respectively) and there is the possibility that the collected outcomes are strictly related to the employed hardware and they cannot be easily generalized to the VR and AR modalities. One possible improvement would be repeating the experiments using only a single device capable of displaying contents using both VR and AR technologies (a possible choice could be the HTC Vive Pro equipped with frontal cameras).

Appendix A

The AR works in the Collaborative Robotic Domain

In the following table, several works regarding the use of the AR interfaces in the HRC domain have been listed and scored. Refer to one of the works done for this Ph.D dissertation [84] for the complete review and for the metric adopted for the evaluation.

Paper	QC1	QC2	QC3	QC4	QC5	Quality
[65]	1	1	0.25	0.79	0.49	3.53
[123]	1	1	0.75	0.24	0.51	3.50
[106]	1	1	0	0.71	0.66	3.37
[294]	1	1	0.25	0.07	1	3.32
[274]	1	1	0	1	0.31	3.31
[105]	1	1	0.25	0.33	0.67	3.25
[323]	1	1	0.58	0.07	0.4	3.05
[330]	1	1	0.25	0.27	0.48	3.00
[321]	1	1	0.25	0.29	0.4	2.94
[437]	1	1	0.25	0.07	0.45	2.77
[90]	1	1	0	0.05	0.63	2.68
[335]	1	1	0	0	0.64	2.64
[337]	0	1	0	0.71	0.72	2.43
[107]	0	1	0.55	0.18	0.69	2.42
[468]	1	1	0	0.12	0.24	2.36
[276]	0	1	0.53	0.5	0.31	2.34
[12]	0	1	0.56	0.5	0.25	2.31
[124]	0	1	0.67	0.29	0.24	2.20
[293]	0	1	0	0.91	0.28	2.19
[352]	0	1	0.79	0.14	0.25	2.18
[136]	0	1	0.57	0	0.61	2.18

[188]	0	1	0.62	0.04	0.39	2.05
[233]	0	1	0.57	0.07	0.39	2.03
[435]	0	1	0.53	0.21	0.25	1.99
[56]	0	1	0	0.39	0.54	1.93
[260]	0	1	0	0.18	0.75	1.93
[249]	0	1	0	0.14	0.78	1.92
[78]	0	1	0.55	0.14	0.19	1.88
[239]	0	1	0.25	0.23	0.39	1.87
[332]	0	1	0.33	0.14	0.36	1.83
[497]	0	1	0.29	0.41	0.1	1.80
[284]	0	1	0.53	0	0.22	1.75
[453]	0	1	0.28	0.06	0.31	1.65
[478]	0	1	0.3	0.07	0.28	1.65
[108]	0	1	0.25	0.09	0.3	1.64
[58]	0	1	0.5	0.03	0.09	1.62
[398]	0	1	0.25	0	0.33	1.58
[36]	0	1	0	0.21	0.36	1.57
[109]	0	1	0	0.2	0.36	1.56
[460]	0	1	0.25	0.18	0.13	1.56
[259]	0	1	0.27	0.07	0.21	1.55
[11]	0	1	0	0.21	0.34	1.55
[261]	0	1	0	0.18	0.36	1.54
[96]	0	1	0	0.06	0.48	1.54
[463]	0	1	0	0.37	0.15	1.52
[365]	0	1	0.25	0.12	0.12	1.49
[462]	0	1	0	0.25	0.24	1.49
[461]	0	1	0.25	0.08	0.16	1.49
[324]	0	1	0	0.15	0.3	1.45
[68]	0	1	0.25	0.01	0.16	1.42
[374]	0	1	0	0.17	0.21	1.38
[238]	0	1	0	0	0.3	1.30
[158]	0	1	0	0.04	0.25	1.29
[277]	0	1	0	0	0.25	1.25
[97]	0	1	0	0	0.19	1.19
[91]	0	1	0	0.04	0.15	1.19
[250]	0	0	0	0.27	0.07	0.34
[459]	0	0	0	0.24	0.07	0.31
[458]	0	0	0	0.14	0.03	0.17
[407]	0	0	0	0.07	0.09	0.16
[15]	0	0	0	0.04	0.1	0.14

[285]	0	0	0	0	0.12	0.12
[14]	0	0	0	0	0.04	0.04

Table A.1: The works related to the use of the AR technology in the HRC context. Interested readers should refer to [84] for the complete assessment review.

Appendix B

The AM Pseudocode

Algorithm 1: The pseudocode of the AM modality.

Input:
pos, orient // *position and orientation of the user*
error_location // *Joint affected by fault position*
joint_values // *Joint orientations*
fault_type
Constant:
r_{min}, r_{max}, D_{min}, D_{max}, D_{robot}, S_c, S_{min}, S_{max}
Start A1:
camera = setCamera(pos, orient)
V = getCameraPosition(camera)
J = getPosJointFault(error_location, joint_values)
D_{JV} = getJointCameraDistance(J, V)
S_{icon} = null
if $D_{JV} < D_{min}$
    $S_{icon} = S_{min}$
else if $D_{JV} >= D_{max}$
    $S_{icon} = S_{max}$
else
    S_{icon} = getScaleIcon(S_c, D_{robot}, D_{JV}) // *see Eq. 2.3*
end if
Start A2:
icon = getIcon(fault_type) // *instantiate the 3D icon*
BB_{3D} = get3DBB(icon)
BB_{2D} = get2DBB(BB_{3D}) // *on the camera plane*

I_{hsv} = getImage(camera)
I_t = getThresholdImage(camera, r_{min}, r_{max})
Q_s = getQuadrantsGrid(I_t, BB_{2D})
Q_{start} = getStartingQuad(Q_s)
Q_{r_max}, Q_{c_max} = getRowColumnQMax(Q_{start}, Q_s)
EL_{max} = max(Q_{r_max}, Q_{c_max})
found_quad = false
while(found_quad == false):
  Q = getQuad(Q_{start}, EL_{max}, k, u) // *see Eq. 2.6 and 2.7*
  if Q is FREE
    $Q_{selected}$ = Q
    found_quad = true
  end if
end while
ray = getRayThroughQ($Q_{selected}$)
R = getPointonRay(ray)
VR = getVector(V, R)
VJ = getVector(V, J)
pos_{3D} = projection(VJ, VR) // *see Fig. 2.18*
Start A3:
look_vector = getLookAtVector(V, pos_{3D})
getIconOrientationFrame(look_vector, icon) // *change icon's orientation*

Appendix C

The Complete Questionnaire

C.1 Questionnaire of Sec. 2.3

Scene 1-2-4	**Response Type**
The robot has performed the whole action without problems	Yes/No
The function of the virtual metaphor was clear and intuitive	1 (totally unclear) 5 (totally clear)
Did you understand the cause of the error?	Yes/No
What was the cause of the error?	
Scene 3	
The robot has performed the whole action without problems	Yes/No
Has a possible dangerous situation occurred?	Yes/No
The function of the virtual metaphor was clear and intuitive	1 (totally unclear) 5 (totally clear)
Did you understand the cause of the interruption?	Yes/No
What was the cause of the interruption?	
In the smartphone version, were the assets more clear?	Yes/No
Did you prefer the AR glasses application or the smartphone version?	AR Glasses Smartphone

Table C.1: The questionnaire used to evaluate the fault metaphors.

C.2 Questionnaire of Sec. 2.4

Category	Question
QR1	How many times have you used an AR application ? How many times have you used an AR HMD? Have you ever worked with a robotic arm ? Do you know the Niryo robot ?
QR2	It was easy to understand which type of fault has occurred. I could access the information at the most appropriate time and place. The icons expressed the most correct amount of information
QR3	The icon was positioned where I could see it. The icon was not positioned where I could see it. The icon had the correct dimension. The icon had not the correct dimension. The icon had the correct orientation. The icon had not the correct orientation.
QR4	The FoV was suitable for the proposed use case.
QR5	I rate this system as

Table C.2: The questionnaire used to evaluate the subjective parameters (scores between 0-4).

C.3 Questionnaire of Sec. 2.5

Q1/Q12	The ITEM have clearly indicated the required real pieces to use during the procedure
Q2/Q13	The animations of the ITEM have clearly shown how to combine the real objects
Q3/Q14	It seemed to me to collaborate with the remote person
Q4/Q15	It seemed to me to work alone.
Q5/Q16	It seemed to me to be in the same room with the remote person.
Q6/Q17	It seemed to me to be alone in the room.
Q7/Q18	The animations and the ITEM have clearly shown how to plug the hand on the end-effector of the robot.
Q8/Q19	I was able to complete the procedures without watching several times the animations.
Q9/Q20	I needed to repeate the procedures several times.
Q10/Q21	The audio instructions have been fundamental to complete the procedure.
Q11/Q22	I think I could complete the procedure without the audio instructions.

Table C.3: Each line represents a question used for both interfaces. The word ITEM should be replaced with *3D arrows* or with *avatar* depending on the questionnaire section.

Nomenclature

Roman Symbols

AGV Automated Guided Vehicle

AR Augmented Reality

BCI Brain Computer Interface

CAD Computer-Aided Design

CRT Cathode Ray Tube

DCT Discrete Cosine Transform

DOF Degrees of Freedom

FoV Field-of-View

HCI Human-Computer Interaction

HMD Head-Mounted Display

HOG Histogram of Oriented Gradients

HRC Human-Robot Collaboration

$HRTF$ Head Related Transfer Function

ICP Iterative Closest Point

$IETM$ Interactive Electronic Technical Manual

ILD Interaural Level Difference

IR Infrared

ISO International Organization for Standardization

ITD Interaural Time Difference

LAN Local Area Network

LCD Liquid Crystal Display

MAR Maintenance-Assembly-Repair

$MEC - SPQ$ MEC Spatial Presence Questionnaire

NUI Natural User Interface

OS Operating System

RGB Red Green Blue

$RGB - D$ Red Green Blue-Depth

ROS Robot Operating System

SDK Software Development Kit

$SIFT$ Scale-Invariant Feature Transform

$SLAM$ Simultaneous Localization and Mapping

SSQ Simulator Sickness Questionnaire

SUS System Usability Scale

TCP Transmission Control Protocol

UDP User Datagram Protocol

UI User Interface

VR Virtual Reality

Bibliography

[1] Lotfi Abdi, Faten Ben Abdallah, and Aref Meddeb. "In-vehicle augmented reality traffic information system: a new type of communication between driver and vehicle". In: *Procedia Computer Science* 73 (2015), pp. 242–249.

[2] Reza Abiri et al. "A comprehensive review of EEG-based brain–computer interface paradigms". In: *Journal of neural engineering* 16.1 (2019), p. 011001.

[3] Tibor Agócs et al. "A large scale interactive holographic display". In: *IEEE Virtual Reality Conference (VR 2006)*. IEEE. 2006, pp. 311–311.

[4] B Akan, B Çürüklü, et al. "Augmented reality meets industry: Interactive robot programming". In: *Proc. SIGRAD*. 052. Västerås: Linköping University Electronic Press, 2010, pp. 55–58.

[5] Ijaz Akhter et al. "Trajectory space: A dual representation for nonrigid structure from motion". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 33.7 (2010), pp. 1442–1456.

[6] Wadee Alhalabi. "Virtual reality systems enhance students' achievements in engineering education". In: *Behaviour & Information Technology* 35.11 (2016), pp. 919–925.

[7] Luis Miguel Alves Fernandes et al. "Exploring educational immersive videogames: an empirical study with a 3D multimodal interaction prototype". In: *Behaviour & Information Technology* 35.11 (2016), pp. 907–918.

[8] Aditya Amberkar et al. "Speech Recognition using Recurrent Neural Networks". In: *2018 International Conference on Current Trends towards Converging Technologies (ICCTCT)*. IEEE. 2018, pp. 1–4.

[9] R. O. Ambrose et al. "Robonaut: NASA's space humanoid". In: *IEEE Intelligent Systems and their Applications* 15.4 (July 2000), pp. 57–63. ISSN: 1094-7167. DOI: 10.1109/5254.867913.

[10] Judith Amores and Pattie Maes. "Essence: Olfactory interfaces for unconscious influence of mood and cognitive performance". In: *Proceedings of the 2017 CHI conference on human factors in computing systems*. 2017, pp. 28–34.

[11] Rasmus S Andersen et al. "Intuitive task programming of stud welding robots for ship construction". In: *2015 IEEE International Conference on Industrial Technology (ICIT)*. IEEE. 2015, pp. 3302–3307.

[12] Rasmus S Andersen et al. "Projecting robot intentions into human environments". In: *2016 25th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*. IEEE. 2016, pp. 294–301.

[13] Christoph Anthes and Jens Volkert. "inVRs – A Framework for Building Interactive Networked Virtual Reality Systems". In: vol. 4208. Sept. 2006, pp. 894–904.

[14] Dejanira Araiza-Illan et al. "Augmented Reality for Quick and Intuitive Robotic Packing Re-Programming". In: *2019 14th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*. IEEE. 2019, pp. 664–664.

[15] Diana Araque et al. "Augmented reality motion-based robotics off-line programming". In: *2011 IEEE Virtual Reality Conference*. IEEE. 2011, pp. 191–192.

[16] Ferran Argelaguet and Morgant Maignant. "GiAnt: stereoscopic-compliant multi-scale navigation in VEs". In: *Proceedings of the 22nd acm conference on virtual reality software and technology*. 2016, pp. 269–277.

[17] K Somani Arun, Thomas S Huang, and Steven D Blostein. "Least-squares fitting of two 3-D point sets". In: *IEEE Transactions on pattern analysis and machine intelligence* 5 (1987), pp. 698–700.

[18] Giancarlo Avalle et al. "An augmented reality system to support fault visualization in industrial robotic tasks". In: *IEEE Access* 7 (2019), pp. 132343–132359.

[19] Benjamin Avery et al. "Evaluation of user satisfaction and learnability for outdoor augmented reality gaming". In: *Proceedings of the 7th Australasian User interface conference-Volume 50*. Australian Computer Society, Inc. 2006, pp. 17–24.

[20] Ronald T Azuma. "A survey of augmented reality". In: *Presence: Teleoperators & Virtual Environments* 6.4 (1997), pp. 355–385.

[21] Huidong Bai et al. "A User Study on Mixed Reality Remote Collaboration with Eye Gaze and Hand Gesture Sharing". In: *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. 2020, pp. 1–13.

[22] Mohd Ali Balafar et al. "Review of brain MRI image segmentation methods". In: *Artificial Intelligence Review* 33.3 (2010), pp. 261–274.

[23] Tibor Balogh, Péter Tamás Kovács, and Attila Barsi. "Holovizio 3D display system". In: *2007 3DTV Conference*. IEEE. 2007, pp. 1–4.

[24] Aaron Bangor, Phil Kortum, and James Miller. "Determining What Individual SUS Scores Mean: Adding an Adjective Rating Scale". In: *J. Usability Stud.* 4 (Apr. 2009), pp. 114–123.

[25] Ali Bashashati et al. "A survey of signal processing algorithms in brain–computer interfaces based on electrical brain signals". In: *Journal of Neural engineering* 4.2 (2007), R32.

[26] Joseph Bates. "Virtual reality, art, and entertainment". In: *Presence: Teleoperators & Virtual Environments* 1.1 (1992), pp. 133–138.

[27] James Baumeister et al. "Cognitive cost of using augmented reality displays". In: *IEEE transactions on visualization and computer graphics* 23.11 (2017), pp. 2378–2388.

[28] Oliver Baus and Stéphane Bouchard. "Exposure to an unpleasant odour increases the sense of presence in virtual reality". In: *Virtual Reality* 21.2 (2017), pp. 59–74.

[29] Johannes Behr and Dieter Fellner. "Instantreality — A Framework for Industrial Augmented and Virtual Reality Applications". In: Jan. 2011, pp. 91–99. ISBN: 978-3-642-17375-2.

[30] D Beymer. "Face recognition under varying pose," in: *Proceedings of 23rd Image Understanding Workshop*. Vol. 2. 1994, pp. 837–842.

[31] Anastasiia Beznosyk et al. "Influence of Network Delay and Jitter on Cooperation in Multiplayer Games". In: (Nov. 2011). DOI: 10.1145/2087756.2087812.

[32] Benoit Bideau et al. "Using virtual reality to analyze sports performance". In: *IEEE Computer Graphics and Applications* 30.2 (2009), pp. 14–21.

[33] Marios Bikos et al. "An interactive augmented reality chess game using bare-hand pinch gestures". In: *2015 International Conference on Cyberworlds (CW)*. IEEE. 2015, pp. 355–358.

[34] Ahmet Birdal and Reza Hassanpour. "Region based hand gesture recognition". In: (2008).

[35] Pierre-Alexandre Blanche et al. "An updatable holographic display for 3D visualization". In: *Journal of display technology* 4.4 (2008), pp. 424–430.

[36] Sebastian Blankemeyer et al. "Intuitive robot programming using augmented reality". In: *Procedia CIRP* 76 (2018), pp. 155–160.

[37] Benjamin Blankertz et al. "Invariant common spatial patterns: Alleviating nonstationarities in brain-computer interfacing". In: *Advances in neural information processing systems*. 2008, pp. 113–120.

[38] Adam Bodnar, Richard Corbett, and Dmitry Nekrasovski. "AROMA: ambient awareness through olfaction in a messaging application". In: *Proceedings of the 6th international conference on Multimodal interfaces*. 2004, pp. 183–190.

[39] Costas Boletsis. "The new era of virtual reality locomotion: A systematic literature review of techniques and a proposed typology". In: *Multimodal Technologies and Interaction* 1.4 (2017), p. 24.

[40] Michael Bonfert et al. "Augmented invaders: a mixed reality multiplayer outdoor game". In: *Proceedings of the 23rd ACM Symposium on Virtual Reality Software and Technology*. ACM. 2017, p. 48.

[41] Wutthigrai Boonsuk. "Evaluation of desktop interface displays for 360-degree video". In: (2011).

[42] Monica Bordegoni et al. "Haptic and sound interface for shape rendering". In: *Presence: Teleoperators and Virtual Environments* 19.4 (2010), pp. 341–363.

[43] Sébastien Bottecchia, Jean-Marc Cieutat, and Jean-Pierre Jessel. "TAC: augmented reality system for collaborative tele-assistance in the field of maintenance through internet". In: *Proceedings of the 1st Augmented Human International Conference*. ACM. 2010, p. 14.

[44] Evren Bozgeyikli et al. "Locomotion in virtual reality for individuals with autism spectrum disorder". In: *Proceedings of the 2016 Symposium on Spatial User Interaction*. 2016, pp. 33–42.

[45] Evren Bozgeyikli et al. "Point & teleport locomotion technique for virtual reality". In: *Proceedings of the 2016 Annual Symposium on Computer-Human Interaction in Play*. 2016, pp. 205–216.

[46] Lars Bretzner, Ivan Laptev, and Tony Lindeberg. "Hand gesture recognition using multi-scale colour features, hierarchical models and particle filtering". In: *Proceedings of fifth IEEE international conference on automatic face gesture recognition*. IEEE. 2002, pp. 423–428.

[47] Wolfgang Broll. "Distributed virtual reality for everyone - a framework for networked VR on the Internet". In: Apr. 1997, pp. 121–128, 217. ISBN: 0-8186-7843-7. DOI: 10.1109/VRAIS.1997.583053.

[48] John Brooke et al. "SUS-A quick and dirty usability scale". In: *Usability evaluation in industry* 189.194 (1996), pp. 4–7.

[49] Martha Burkle and Michael Magee. "Virtual learning: videogames and virtual reality in education". In: *Digital Tools for Seamless Learning*. IGI Global, 2017, pp. 325–344.

[50] Caroline Bushdid et al. “Humans can discriminate more than 1 trillion olfactory stimuli”. In: *Science* 343.6177 (2014), pp. 1370–1372.

[51] Marcio C Cabral, Carlos H Morimoto, and Marcelo K Zuffo. “On the usability of gesture interfaces in virtual reality environments”. In: *Proceedings of the 2005 Latin American conference on Human-computer interaction*. 2005, pp. 100–108.

[52] Chao Cao, Marius Preda, and Titus Zaharia. “3D point cloud compression: A survey”. In: *The 24th International Conference on 3D Web Technology*. 2019, pp. 1–9.

[53] Jorge CS Cardoso. “Comparison of gesture, gamepad, and gaze-based locomotion for VR worlds”. In: *Proceedings of the 22nd ACM conference on virtual reality software and technology*. 2016, pp. 319–320.

[54] G. S. Carson, R. F. Puk, and R. Carey. “Developing the VRML 97 international standard”. In: *IEEE Computer Graphics and Applications* 19.2 (1999), pp. 52–58.

[55] Ravi Teja Chadalavada et al. “That’s on my mind! robot to human intention communication through on-board projection on shared floor space”. In: *Proc. ECMR*. Lincoln: IEEE, 2015, pp. 1–6.

[56] Tathagata Chakraborti et al. “Projection-aware task planning and execution for human-in-the-loop operation of robots in a mixed-reality workspace”. In: *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2018, pp. 4476–4482.

[57] Biplab Ketan Chakraborty et al. “Review of constraints on vision-based gesture recognition for human–computer interaction”. In: *IET Computer Vision* 12.1 (2017), pp. 3–15.

[58] Siam Charoenseang and Tarinee Tonggoed. “Human–robot collaboration with augmented reality”. In: *International Conference on Human-Computer Interaction*. Springer. 2011, pp. 93–97.

[59] Pierre Chastenay. “From geocentrism to allocentrism: Teaching the phases of the moon in a digital full-dome planetarium”. In: *Research in Science Education* 46.1 (2016), pp. 43–77.

[60] Jie Chen and Ron J Patton. *Robust model-based fault diagnosis for dynamic systems*. Vol. 3. Springer Science & Business Media, 2012.

[61] Lingchen Chen et al. “A survey on hand gesture recognition”. In: *2013 International conference on computer sciences and applications*. IEEE. 2013, pp. 313–316.

[62] Yen-Wei Chen and Kenji Kubo. "A robust eye detection and tracking technique using gabor filters". In: *Third International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP 2007)*. Vol. 1. IEEE. 2007, pp. 109–112.

[63] Adrian David Cheok et al. "Human Pacman: a mobile, wide-area entertainment system based on physical, social, and ubiquitous computing". In: *Personal and ubiquitous computing* 8.2 (2004), pp. 71–81.

[64] Luca Chittaro and Fabio Buttussi. "Assessing knowledge retention of an immersive serious game vs. a traditional education method in aviation safety". In: *IEEE transactions on visualization and computer graphics* 21.4 (2015), pp. 529–538.

[65] Jonathan Wun Shiung Chong et al. "Robot programming using augmented reality: An interactive method for planning collision-free paths". In: *Robotics and Computer-Integrated Manufacturing* 25.3 (2009), pp. 689–701.

[66] Jesper Vang Christensen et al. "Player Experience in a VR and Non-VR Multiplayer Game". In: *Proceedings of the Virtual Reality International Conference-Laval Virtual*. ACM. 2018, p. 10.

[67] Kyung Ho Chung. "Application of augmented reality to dimensional and geometric inspection". PhD thesis. Blacksburg: Virginia Tech, 2002.

[68] Lennart Claassen et al. "Intuitive Robot Control with a Projected Touch Interface". In: *International Conference on Social Robotics*. Springer. 2014, pp. 95–104.

[69] Jeremy Clifton and Stephen Palmisano. "Effects of steering locomotion and teleporting on cybersickness and presence in HMD-based virtual reality". In: *Virtual Reality* 24.3 (2020), pp. 453–468.

[70] Jacob Cohen. *Statistical power analysis for the behavioral sciences*. Academic press, 2013.

[71] Timothy F. Cootes, Gareth J. Edwards, and Christopher J. Taylor. "Active appearance models". In: *IEEE Transactions on pattern analysis and machine intelligence* 23.6 (2001), pp. 681–685.

[72] Diogo Cordeiro, Nuno Correia, and Rui Jesus. "ARZombie: A mobile augmented reality game with multimodal interaction". In: *2015 7th International Conference on Intelligent Technologies for Interactive Entertainment (INTETAIN)*. IEEE. 2015, pp. 22–31.

[73] Alexandra Covaci et al. "Is multimedia multisensorial?-a review of mulsemedia systems". In: *ACM Computing Surveys (CSUR)* 51.5 (2018), pp. 1–35.

[74] Mario Covarrubias et al. "Flexible touch sensor for evaluating geometric properties of virtual shapes through sound: This paper reports a sonification approach to visualise geometric features that are missing in haptic display". In: *Virtual and Physical Prototyping* 10.2 (2015), pp. 77–89.

[75] Antonio Criminisi, Jamie Shotton, and Ender Konukoglu. "Decision forests for classification, regression, density estimation, manifold learning and semi-supervised learning". In: *Microsoft Research Cambridge, Tech. Rep. MSRTR-2011-114* 5.6 (2011), p. 12.

[76] Alberto Cruz and Barry G Green. "Thermal stimulation of taste". In: *Nature* 403.6772 (2000), pp. 889–892.

[77] Heather Culbertson and Katherine J Kuchenbecker. "Importance of matching physical friction, hardness, and texture in creating realistic haptic virtual surfaces". In: *IEEE transactions on haptics* 10.1 (2016), pp. 63–74.

[78] Oscar Danielsson et al. "Assessing instructions in augmented reality for human-robot collaborative assembly by using demonstrators". In: *Procedia CIRP* 63 (2017), pp. 89–94.

[79] Nasser H Dardas and Nicolas D Georganas. "Real-time hand gesture detection and recognition using bag-of-features and support vector machine techniques". In: *IEEE Transactions on Instrumentation and measurement* 60.11 (2011), pp. 3592–3607.

[80] Dragoş Datcu et al. "Virtual co-location to support remote assistance for inflight maintenance in ground training for space missions". In: *Proc. CompSysTech*. New York: ACM, 2014, pp. 134–141.

[81] Francesca De Crescenzio et al. "Augmented reality for aircraft maintenance training and operations support". In: *IEEE Computer Graphics and Applications* 31.1 (Dec. 2011), pp. 96–101. DOI: 10.1109/MCG.2011.4.

[82] Francesco De Pace, Federico Manuri, and Andrea Sanna. "Augmented reality in industry 4.0". In: *American Journal of Computer Science and Information Technology* 6.1 (2018), p. 17.

[83] Francesco De Pace et al. "A comparison between two different approaches for a collaborative mixed-virtual environment in industrial maintenance". In: *Frontiers in Robotics and AI* 6 (2019), p. 18.

[84] Francesco De Pace et al. "A systematic review of Augmented Reality interfaces for collaborative industrial robots". In: *Computers & Industrial Engineering* 149 (2020), p. 106806.

[85] Francesco De Pace et al. "An augmented interface to display industrial robot faults". In: *International Conference on Augmented Reality, Virtual Reality and Computer Graphics*. Springer. 2018, pp. 403–421.

[86] Francesco De Pace et al. “An Evaluation of Game Usability in Shared Mixed and Virtual Environments”. In: *Game Design and Intelligent Interaction*. IntechOpen, 2019.

[87] Francesco De Pace et al. “Assessing the Suitability and Effectiveness of Mixed Reality Interfaces for Accurate Robot Teleoperation”. In: *26th ACM Symposium on Virtual Reality Software and Technology*. 2020, pp. 1–3.

[88] Francesco De Pace et al. “Virtual and augmented reality interfaces in shared game environments: A novel approach”. In: *International Conference on Intelligent Technologies for Interactive Entertainment*. Springer. 2018, pp. 137–147.

[89] Lucio T De Paolis, Marco Pulimeno, and Giovanni Aloisio. “DIFFERENT SIMULATIONS OF A BILLIARDS GAME.” In: *Journal of Applied Quantitative Methods* 4.4 (2009).

[90] Davide De Tommaso, Sylvain Calinon, and Darwin G Caldwell. “A tangible interface for transferring skills”. In: *International Journal of Social Robotics* 4.4 (2012), pp. 397–408.

[91] Crina Narcisa Deac et al. “Using Augmented Reality in Smart Manufacturing”. In: *Proceedings of the 28th DAAAM International Symposium*. 2017, pp. 0727–0732.

[92] Shuchisnigdha Deb, Lesley J Strawderman, and Daniel W Carruth. “Investigating pedestrian suggestions for external features on fully autonomous vehicles: A virtual reality experiment”. In: *Transportation research part F: traffic psychology and behaviour* 59 (2018), pp. 135–149.

[93] Jeannine Delwiche. “Are there ‘basic’tastes?” In: *Trends in Food Science & Technology* 7.12 (1996), pp. 411–415.

[94] D Derbyshire. “The headset that will mimic all five senses and make the virtual world as convincing as real life”. In: *Web. URL http://www. dailymail. co. uk/sciencetech/article-1159206/The-headset-mimic-senses-make-virtual-worldconvincing-real-life. html* (2009).

[95] Niladri Sekhar Dey, Ramakanta Mohanty, and KL Chugh. “Speech and speaker recognition system using artificial neural networks and hidden markov model”. In: *2012 International Conference on Communication Systems and Network Technologies*. IEEE. 2012, pp. 311–315.

[96] Andre Dietrich et al. “Visualization of robot’s awareness and perception”. In: *Proceedings of the First International Workshop on Digital Engineering*. ACM. 2010, pp. 38–44.

[97] Huy Dinh et al. “Augmented reality interface for taping robot”. In: *2017 18th International Conference on Advanced Robotics (ICAR)*. IEEE. 2017, pp. 275–280.

[98] Fei Dong et al. “Speech Emotion Recognition Based on Multi-Output GMM and SVM”. In: *2010 Chinese Conference on Pattern Recognition (CCPR)*. IEEE. 2010, pp. 1–4.

[99] KD Eason, L Damodaran, TFM Stewart, et al. “Serving the naive computer user”. In: *Omega* 3.3 (1975), pp. 352–353.

[100] Fouzia El Mountassir et al. “Encoding odorant mixtures by human olfactory receptors”. In: *Flavour and Fragrance Journal* 31.5 (2016), pp. 400–407.

[101] G Erboz. “How to Define Industry 4.0: The Main Pillars of Industry 4.0. 2017”. In: ().

[102] Henrik Eschen et al. “Augmented and virtual reality for inspection and maintenance processes in the aviation industry”. In: *Procedia manufacturing* 19 (2018), pp. 156–163.

[103] A Evlampev and M Ostanin. “Obstacle avoidance for robotic manipulator using Mixed reality glasses”. In: *2019 3rd School on Dynamics of Complex Networks and their Application in Intellectual Robotics (DCNAIR)*. IEEE. 2019, pp. 46–48.

[104] Georg E Fabiani et al. “Conversion of EEG activity into cursor movement by a brain-computer interface (BCI)”. In: *IEEE transactions on neural systems and rehabilitation engineering* 12.3 (2004), pp. 331–338.

[105] HC Fang, SK Ong, and AYC Nee. “A novel augmented reality-based interface for robot path planning”. In: *International Journal on Interactive Design and Manufacturing (IJIDeM)* 8.1 (2014), pp. 33–42.

[106] HC Fang, SK Ong, and AYC Nee. “Interactive robot trajectory planning and simulation using augmented reality”. In: *Robotics and Computer-Integrated Manufacturing* 28.2 (2012), pp. 227–237.

[107] HC Fang, SK Ong, and AYC Nee. “Novel AR-based interface for human-robot interaction and visualization”. In: *Advances in Manufacturing* 2.4 (2014), pp. 275–288.

[108] HC Fang, SK Ong, and AYC Nee. “Robot path and end-effector orientation planning using augmented reality”. In: *Procedia CIRP* 3 (2012), pp. 191–196.

[109] Hongchao Fang, Soh Khim Ong, and Andrew Yeh-Ching Nee. “Robot programming using augmented reality”. In: *2009 International Conference on CyberWorlds*. IEEE. 2009, pp. 13–20.

[110] Cesare Fantuzzi, Cristian Secchi, and Antonio Visioli. “On the fault detection and isolation of industrial robot manipulators”. In: *IFAC Proceedings Volumes* 36.17 (2003), pp. 399–404.

[111] Yasin Farmani and Robert J Teather. "Viewpoint snapping to reduce cybersickness in virtual reality". In: *Proceedings of the 44th Graphics Interface Conference*. 2018, pp. 168–175.

[112] Simon Fear. *Publication quality tables in LATEX*. 2005.

[113] Steven Feiner, Blair MacIntyre, and Doree Seligmann. "Knowledge-based augmented reality". In: *Communications of the ACM* 36.7 (1993), pp. 53–62.

[114] Steven Feiner et al. "A touring machine: Prototyping 3D mobile augmented reality systems for exploring the urban environment". In: *Personal Technologies* 1.4 (1997), pp. 208–217.

[115] Andreas Fender, Jorg Muller, and David Lindlbauer. "Creature teacher: A performance-based animation system for creating cyclic movements". In: *Proceedings of the 3rd ACM Symposium on Spatial User Interaction*. 2015, pp. 113–122.

[116] Kai-ping Feng and Fang Yuan. "Static hand gesture recognition based on HOG characters and support vector machines". In: *2013 2nd International Symposium on Instrumentation and Measurement, Sensor Network and Automation (IMSNA)*. IEEE. 2013, pp. 936–938.

[117] Peter Ferdinand et al. "The Eduventure–A new approach of digital game based learning combining virtual and mobile augmented reality games episodes". In: *Pre-Conference Workshop "Game based Learning" of DeLFI 2005 and GMW 2005 Conference, Rostock*. Vol. 13. 2005.

[118] Andrea Ferracani et al. "Locomotion by natural gestures for immersive virtual environments". In: *Proceedings of the 1st international workshop on multimedia alternate realities*. 2016, pp. 21–24.

[119] George W Fitzmaurice. "Situated information spaces and spatially aware palmtop computers". In: *Communications of the ACM* 36.7 (1993), pp. 39–49.

[120] Edward Forgey. "Cluster analysis of multivariate data: Efficiency vs. interpretability of classification". In: *Biometrics* 21.3 (1965), pp. 768–769.

[121] Eric Foxlin. "Inertial head-tracker sensor fusion by a complementary separate-bias Kalman filter". In: *Proceedings of the IEEE 1996 Virtual Reality Annual International Symposium*. IEEE. 1996, pp. 185–194.

[122] Alexandre R Francois. *Real-time multi-resolution blob tracking*. Tech. rep. UNIVERSITY OF SOUTHERN CALIFORNIA LOS ANGELES INST FOR ROBOTICS AND ..., 2004.

[123] Jared A Frank, Matthew Moorhead, and Vikram Kapila. "Mobile mixed-reality interfaces that enhance human–robot interaction in shared spaces". In: *Frontiers in Robotics and AI* 4 (2017), pp. 1–20.

[124] Jared A Frank, Matthew Moorhead, and Vikram Kapila. "Realizing mixed-reality environments with tablets for intuitive human-robot collaboration for object manipulation tasks". In: *2016 25th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*. IEEE. 2016, pp. 302–307.

[125] Anton Franzluebbers and Kyle Johnson. "Remote Robotic Arm Teleoperation through Virtual Reality". In: *Symposium on Spatial User Interaction*. 2019, pp. 1–2.

[126] Johannes Frasnelli et al. "Odor localization and sniffing". In: *Chemical senses* 34.2 (2009), pp. 139–144.

[127] Christopher Frauenberger and Markus Noistering. "3D audio interfaces for the blind". In: Georgia Institute of Technology. 2003.

[128] Joakim Grant Frederiksen et al. "Cognitive load and performance in immersive virtual reality versus conventional virtual reality simulation training of laparoscopic surgery: a randomized trial". In: *Surgical endoscopy* 34.3 (2020), pp. 1244–1252.

[129] Laura Freina and Michela Ott. "A literature review on immersive virtual reality in education: state of the art and perspectives". In: *The international scientific conference elearning and software for education*. Vol. 1. 133. 2015, pp. 10–1007.

[130] Joseph J Fuller and Carderock Div. "IETMs: From research to reality". In: *CALS Expo'94 International Proceedings* (1994).

[131] Santosh K Gaikwad, Bharti W Gawali, and Pravin Yannawar. "A review on speech recognition technique". In: *International Journal of Computer Applications* 10.3 (2010), pp. 16–24.

[132] Alberto Gallace et al. "Using a small size olfactory device to affect people's taste of food: preliminary evidence". In: *Proceedings of the Mul- tisensory Human Computer Interaction (CHI) 2016*. 2016.

[133] Akemi Gálvez and Andrés Iglesias. "Videogames and virtual reality as effective edutainment tools". In: *International Conference on Future Generation Information Technology*. Springer. 2010, pp. 564–576.

[134] Luigi Gammieri et al. "Coupling of a redundant manipulator with a virtual reality environment to enhance human-robot cooperation". In: *Procedia CIRP* 62 (2017), pp. 618–623.

[135] Xin Gao et al. “High brightness three-dimensional light field display based on the aspheric substrate Fresnel-lens-array with eccentric pupils”. In: *Optics Communications* 361 (2016), pp. 47–54.

[136] Yuxiang Gao and Chien-Ming Huang. “PATI: a projection-based augmented table-top interface for robot programming”. In: *Proceedings of the 24th International Conference on Intelligent User Interfaces*. ACM. 2019, pp. 345–355.

[137] Augusto Garcia-Agundez et al. “Development of a classifier to determine factors causing cybersickness in virtual reality environments”. In: *Games for health journal* 8.6 (2019), pp. 439–444.

[138] S. Garrido-Jurado et al. “Automatic generation and detection of highly reliable fiducial markers under occlusion”. In: *Pattern Recognition* 47.6 (2014), pp. 2280–2292.

[139] Sanket Gaurav et al. “Deep Correspondence Learning for Effective Robotic Teleoperation using Virtual Reality”. In: *2019 IEEE-RAS 19th International Conference on Humanoid Robots (Humanoids)*. IEEE. 2019, pp. 477–483.

[140] Andrew Gee and Roberto Cipolla. “Determining the gaze of faces in images”. In: *Image and Vision Computing* 12.10 (1994), pp. 639–647.

[141] Pierre Georgel et al. “An industrial augmented reality solution for discrepancy check”. In: *2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality*. IEEE. 2007, pp. 111–115.

[142] Wiqas Ghai and Navdeep Singh. “Literature review on automatic speech recognition”. In: *International Journal of Computer Applications* 41.8 (2012).

[143] Warren Giles, Ralph Schroeder, and Bryan Cleal. “Virtual reality and the future of interactive games”. In: *Virtual Reality’94*. Springer, 1994, pp. 377–391.

[144] Toni Giorgino. “Computing and Visualizing Dynamic Time Warping Alignments in R: The dtw Package”. In: *Journal of Statistical Software* 31.7 (2009), pp. 1–24. DOI: 10.18637/jss.v031.i07.

[145] James Gips et al. “Using EagleEyes—an electrodes based device for controlling the computer with your eyes—to help people with special needs”. In: *Proceedings of the 5th International conference on Computers helping people with special needs. Part I*. 1996, pp. 77–83.

[146] Mani Golparvar-Fard, F Peña-Mora, and S Savarese. “D4AR–a 4-dimensional augmented reality model for automating construction progress monitoring data collection, processing and communication”. In: *Journal of information technology in construction* 14.13 (2009), pp. 129–153.

[147] Mani Golparvar-Fard et al. "Visualization of construction progress monitoring with 4D simulation model overlaid on time-lapsed photographs". In: *Journal of Computing in Civil Engineering* 23.6 (Nov. 2009), pp. 391–404. DOI: 10.1061/(ASCE)0887-3801(2009)23:6(391).

[148] Jun Gong, Peter Tarasewich, et al. "Guidelines for handheld mobile device interface design". In: *Proceedings of DSI 2004 Annual Meeting*. Citeseer. 2004, pp. 3751–3756.

[149] Andrés Vargas González et al. "A comparison of desktop and augmented reality scenario based training authoring tools". In: *2019 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*. IEEE. 2019, pp. 339–350.

[150] Stuart Goose and Carsten Möller. "A 3D audio only interactive Web browser: using spatialization to convey hypermedia document structure". In: *Proceedings of the seventh ACM international conference on Multimedia (Part 1)*. 1999, pp. 363–371.

[151] Stuart Goose et al. "Speech-enabled augmented reality supporting mobile industrial maintenance". In: *IEEE Pervasive Computing* 2.1 (Jan. 2003), pp. 65–70. DOI: 10.1109/MPRV.2003.1186727.

[152] Michihiko Goto et al. "Task support system by displaying instructional video onto AR workspace". In: *Mixed and Augmented Reality (ISMAR), 2010 9th IEEE International Symposium on*. IEEE. 2010, pp. 83–90.

[153] Mahesh Goyani, Namrata Dave, and NM Patel. "Performance analysis of lip synchronization using LPC, MFCC and PLP speech parameters". In: *2010 international conference on computational intelligence and communication networks*. IEEE. 2010, pp. 582–587.

[154] Jerônimo G Grandi et al. "Design and assessment of a collaborative 3D interaction technique for handheld augmented reality". In: *2018 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*. IEEE. 2018, pp. 49–56.

[155] Enrico Gregorio. "Installing TeX Live 2010 on Ubuntu". In: *TUGboat* 32.1 (2011), pp. 56–61.

[156] W Andrew Grubbs. *Telerobotic surgery system for remote surgeon training using remote surgery station and party conferencing and associated methods*. US Patent App. 15/138,427. 2016.

[157] Zebiao Guan et al. "A novel robot teaching system based on augmented reality". In: *2019 International Conference on Image and Video Processing, and Artificial Intelligence*. Vol. 11321. International Society for Optics and Photonics. 2019, p. 113211D.

[158] Jan Guhl, Johannes Hügle, and Jörg Krüger. "Enabling Human-Robot-Interaction via Virtual and Augmented Reality in Distributed Control Systems". In: *Procedia CIRP* 76 (2018), pp. 167–170.

[159] Ziyue Guo et al. "Using virtual reality to support the product's maintainability design: Immersive maintainability verification and evaluation system". In: *Computers in Industry* 101 (2018), pp. 41–50.

[160] Kartiki Gupta and Divya Gupta. "An analysis on LPC, RASTA and MFCC techniques in Automatic Speech recognition system". In: *2016 6th International Conference-Cloud System and Big Data Engineering (Confluence)*. IEEE. 2016, pp. 493–497.

[161] Martin Hachet, Joachim Pouderoux, and Pascal Guitton. "A camera-based interface for interaction with mobile handheld computers". In: *Proceedings of the 2005 symposium on Interactive 3D graphics and games*. 2005, pp. 65–72.

[162] Jonna Häkkilä et al. "Visiting a virtual graveyard: designing virtual reality cultural heritage experiences". In: *Proceedings of the 18th International Conference on Mobile and Ubiquitous Multimedia*. 2019, pp. 1–4.

[163] Dan Witzner Hansen and Qiang Ji. "In the eye of the beholder: A survey of models for eyes and gaze". In: *IEEE transactions on pattern analysis and machine intelligence* 32.3 (2009), pp. 478–500.

[164] Peter RG Harding and Timothy Ellis. "Recognizing hand gesture using Fourier descriptors". In: *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.* Vol. 3. IEEE. 2004, pp. 286–289.

[165] Surina Hariri et al. "Electrical stimulation of olfactory receptors for digitizing smell". In: *Proceedings of the 2016 workshop on Multimodal Virtual and Augmented Reality*. 2016, pp. 1–4.

[166] Tom Haritos and Nickolas D Macchiarella. "A mobile application of augmented reality for aerospace maintenance training". In: *24th digital avionics systems conference*. Vol. 1. IEEE. 2005, 5–B.

[167] Panu Harmo et al. "Etala-virtual reality assisted telepresence system for remote control and maintenance". In: *IFAC Proceedings Volumes* 33.26 (2000), pp. 1011–1016.

[168] Christopher G Harris, Mike Stephens, et al. "A combined corner and edge detector." In: *Alvey vision conference*. Vol. 15. 50. Citeseer. 1988, pp. 10–5244.

[169] Sandra G Hart and Lowell E Staveland. "Development of NASA-TLX (Task Load Index): Results of empirical and theoretical research". In: *Advances in psychology*. Vol. 52. Elsevier, 1988, pp. 139–183.

[170] Haitham Hasan and S Abdul-Kareem. "RETRACTED ARTICLE: Static hand gesture recognition using neural networks". In: *Artificial Intelligence Review* 41.2 (2014), pp. 147–181.

[171] Mokhtar M Hasan and Pramod K Mishra. "Hand gesture modeling and recognition using geometric features: a review". In: *Canadian journal on image processing and computer vision* 3.1 (2012), pp. 12–26.

[172] Marc Hassenzahl, Michael Burmester, and Franz Koller. "AttrakDiff: Ein Fragebogen zur Messung wahrgenommener hedonischer und pragmatischer Qualität". In: *Mensch & computer 2003*. Springer, 2003, pp. 187–196.

[173] Morton L Heilig. *Sensorama simulator*. US Patent 3,050,870. Aug. 1962.

[174] Steven Henderson and Steven Feiner. "Exploring the benefits of augmented reality documentation for maintenance and repair". In: *IEEE transactions on visualization and computer graphics* 17.10 (2010), pp. 1355–1368.

[175] Steven J Henderson and Steven K Feiner. *Augmented reality for maintenance and repair (armar)*. Tech. rep. Columbia Univ New York Dept of Computer Science, 2007.

[176] Juan David Hernández et al. "Increasing robot autonomy via motion planning and an augmented reality interface". In: *IEEE Robotics and Automation Letters* 5.2 (2020), pp. 1017–1023.

[177] Nicolas S Herrera and Ryan P McMahan. "Development of a simple and low-cost olfactory display for immersive media experiences". In: *Proceedings of the 2nd ACM international workshop on immersive media experiences*. 2014, pp. 1–6.

[178] Peter Hertel. "Writing Articles with LATEX". In: (2010).

[179] Stephen L Hicks et al. "A depth-based head-mounted visual display to aid navigation in partially sighted individuals". In: *PloS one* 8.7 (2013), e67695.

[180] Ryuji Hirayama et al. "A volumetric display for visual, tactile and audio presentation using acoustic trapping". In: *Nature* 575.7782 (2019), pp. 320–323.

[181] Deborah Hix and H Rex Hartson. *Developing user interfaces: Ensuring usability through product and process*. Wiley, 1993.

[182] Erik Hofmann and Marco Rüsch. "Industry 4.0 and the current status as well as future prospects on logistics". In: *Computers in industry* 89 (2017), pp. 23–34.

[183] Radovan Holubek, Roman Ruzarovsky, and Daynier Rolando Delgado Sobrino. "Using virtual reality as a support tool for the offline robot programming". In: *Research Papers Faculty of Materials Science and Technology Slovak University of Technology* 26.42 (2018), pp. 85–91.

[184] Matthias Hoppe et al. "VRHapticDrones: Providing haptics in virtual reality through quadcopters". In: *Proceedings of the 17th International Conference on Mobile and Ubiquitous Multimedia*. 2018, pp. 7–18.

[185] Thanarat Horprasert, Yaser Yacoob, and Larry S Davis. "Computing 3-d head orientation from a monocular image sequence". In: *Proceedings of the second international conference on automatic face and gesture recognition*. IEEE. 1996, pp. 242–247.

[186] Nikola Horvat et al. "Comparing virtual reality and desktop interface for reviewing 3D CAD models". In: *Proceedings of the Design Society: International Conference on Engineering Design*. Vol. 1. 1. Cambridge University Press. 2019, pp. 1923–1932.

[187] Jie Huang et al. "Robotic spatial sound localization and its 3D sound human interface". In: *First International Symposium on Cyber Worlds, 2002. Proceedings*. IEEE. 2002, pp. 191–197.

[188] Johannes Hügle, Jens Lambrecht, and Jörg Krüger. "An integrated approach for industrial robot control and programming combining haptic and non-haptic gestures". In: *2017 26th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*. IEEE. 2017, pp. 851–857.

[189] King-Chu Hung. "The generalized uniqueness wavelet descriptor for planar closed curves". In: *IEEE Transactions on image processing* 9.5 (2000), pp. 834–845.

[190] Wijnand A IJsselsteijn, Yvonne AW de Kort, and Karolien Poels. "The game experience questionnaire". In: *Eindhoven: Technische Universiteit Eindhoven* 46.1 (2013).

[191] Takeshi Imai, Hitoshi Sakano, and Leslie B Vosshall. "Topographic mapping—the olfactory system". In: *Cold Spring Harbor perspectives in biology* 2.8 (2010), a001776.

[192] Masaharu Inoue et al. "A novel approach to patient self-monitoring of sonographic examinations using a head-mounted display". In: *Journal of Ultrasound in Medicine* 34.1 (2015), pp. 29–35.

[193] iso.com. *ISO 10218-1:2011*. URL: https://www.iso.org/standard/51330.html.

[194] iso.com. *ISO 10218-2:2011*. URL: https://www.iso.org/standard/41571.html.

[195] iso.com. *ISO 15066:2016*. URL: https://www.iso.org/standard/62996.html.

[196] iso.com. *ISO 8373:2012(en) Robots and robotic devices — Vocabulary*. URL: https://www.iso.org/obp/ui/#iso:std:iso:8373:ed-2:v1:en.

[197] iso.com. *SO 9241-210:2019 Ergonomics of human-system interaction — Part 210: Human-centred design for interactive systems*. URL: https://www.iso.org/standard/77520.html.

[198] Shahram Izadi et al. "KinectFusion: Real-time 3D reconstruction and interaction using a moving depth camera". In: Oct. 2011, pp. 559–568. DOI: 10.1145/2047196.2047270.

[199] Changwon Jang et al. "Retinal 3D: augmented reality near-eye display via pupil-tracked light field projection on retina". In: *ACM Transactions on Graphics (TOG)* 36.6 (2017), pp. 1–13.

[200] Dongsik Jo, Ki-Hong Kim, and Gerard Jounghyun Kim. "Effects of avatar and background types on users' co-presence and trust for mixed reality-based teleconference systems". In: *In Proceedings the 30th Conference on Computer Animation and Social Agents*. 2017, pp. 27–36.

[201] Dongsik Jo, Ki-Hong Kim, and Gerard Jounghyun Kim. "SpaceTime: adaptive control of the teleported avatar for improved AR tele-conference experience". In: *Computer Animation and Virtual Worlds* 26.3-4 (2015), pp. 259–269.

[202] Magnus Johnsson and Christian Balkenius. "Sense of touch in robots with self-organizing maps". In: *IEEE Transactions on Robotics* 27.3 (2011), pp. 498–507.

[203] Bhuvaneshwari Jolad and Rajashri Khanai. "An Art of Speech Recognition: A Review". In: *2019 2nd International Conference on Signal Processing and Communication (ICSPC)*. IEEE. 2019, pp. 31–35.

[204] Marie Jonsson, Clifford Nass, and Kwan Min Lee. "Mixing personal computer and handheld interfaces and devices: effects on perceptions and attitudes". In: *International Journal of Human-Computer Studies* 61.1 (2004), pp. 71–83.

[205] Jean-Marc Jot, Veronique Larcher, and Olivier Warusfel. "Digital signal processing issues in the context of binaural and transaural stereophony". In: *Audio Engineering Society Convention 98*. Audio Engineering Society. 1995.

[206] Shanon X Ju et al. "Analysis of gesture and action in technical talks for video indexing". In: *Proceedings of IEEE computer society conference on computer vision and pattern recognition*. IEEE. 1997, pp. 595–601.

[207] Mohamed Kaâniche. "Gesture recognition from video sequences". PhD thesis. 2009.

[208] Henning Kagermann, Wolfgang Wahlster, Johannes Helbig, et al. "Recommendations for implementing the strategic initiative Industrie 4.0: Final report of the Industrie 4.0 Working Group". In: *Forschungsunion: Berlin, Germany* (2013).

[209] Svenja Kahn et al. "3d discrepancy check via augmented reality". In: *Proc. ISMAR*. Seoul: IEEE, 2010, pp. 241–242. DOI: 10.1109/ISMAR.2010.5643587.

[210] Svenja Kahn et al. "Beyond 3d" as-built" information using mobile ar enhancing the building lifecycle management". In: *2012 International Conference on Cyberworlds*. IEEE. 2012, pp. 29–36.

[211] Hirokazu Kato and Mark Billinghurst. "Marker tracking and hmd calibration for a video-based augmented reality conferencing system". In: *Proceedings 2nd IEEE and ACM International Workshop on Augmented Reality (IWAR'99)*. IEEE. 1999, pp. 85–94.

[212] Shinjiro Kawato and Nobuji Tetsutani. "Detection and tracking of eyes for gaze-camera control". In: *Image and Vision Computing* 22.12 (2004), pp. 1031–1038.

[213] Shinjiro Kawato and Nobuji Tetsutani. "Real-time detection of between-the-eyes with a circle frequency filter". In: *Proceedings of The 5th asian conference on computer vision (ACCV2002)*. Vol. 2. 2002, pp. 442–447.

[214] Shinji Kawatsuma, Mineo Fukushima, and Takashi Okada. "Emergency response by robots to Fukushima-Daiichi accident: summary and lessons learned". In: *Industrial Robot: An International Journal* 39.5 (2012), pp. 428–435.

[215] Joseph Nathaniel Kaye. "Symbolic olfactory display". PhD thesis. Massachusetts Institute of Technology, 2001.

[216] John Kelso et al. "DIVERSE: A Framework for Building Extensible and Reconfigurable Device Independent Virtual Environments". In: Feb. 2002, pp. 183–190. ISBN: 0-7695-1492-8. DOI: 10.1109/VR.2002.996521.

[217] Robert S Kennedy et al. "Simulator sickness questionnaire: An enhanced method for quantifying simulator sickness". In: *The international journal of aviation psychology* 3.3 (1993), pp. 203–220.

[218] Cem Keskin et al. "Real time hand pose estimation using depth sensors". In: *Consumer depth cameras for computer vision*. Springer, 2013, pp. 119–137.

[219] Ng Yong Yi Kevin, S Ranganath, and D Ghosh. "Trajectory modeling in gesture recognition using CyberGloves/sup/spl reg//and magnetic trackers". In: *2004 IEEE Region 10 Conference TENCON 2004*. IEEE. 2004, pp. 571–574.

[220] A Khorshidtalab and Momoh Jimoh Emiyoka Salami. "EEG signal classification for real-time brain-computer interface applications: A review". In: *2011 4th International Conference on Mechatronics (ICOM)*. IEEE. 2011, pp. 1–7.

[221] Seungwon Kim, Gun A Lee, and Nobuchika Sakata. "Comparing pointing and drawing for remote collaboration". In: *Mixed and Augmented Reality (ISMAR), 2013 IEEE International Symposium on*. IEEE. 2013, pp. 1–6.

[222] Soochan Kim et al. "Head mouse system based on gyro-and opto-sensors". In: *2010 3rd International Conference on Biomedical Engineering and Informatics*. Vol. 4. IEEE. 2010, pp. 1503–1506.

[223] Sunjun Kim et al. "iLight: information flashlight on objects using handheld projector". In: *CHI'10 Extended Abstracts on Human Factors in Computing Systems*. 2010, pp. 3631–3636.

[224] LM King, HT Nguyen, and PB Taylor. "Hands-free head-movement gesture recognition using artificial neural networks and the magnified gradient function". In: *2005 IEEE Engineering in Medicine and Biology 27th Annual Conference*. IEEE. 2006, pp. 2063–2066.

[225] Naohiro Kishishita et al. "Analysing the effects of a wide field of view augmented reality display on search performance in divided attention tasks". In: *2014 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*. IEEE. 2014, pp. 177–186.

[226] Alexandra Kitson et al. "Comparing leaning-based motion cueing interfaces for virtual reality locomotion". In: *2017 IEEE Symposium on 3d user interfaces (3DUI)*. IEEE. 2017, pp. 73–82.

[227] Georg Klein and David Murray. "Parallel tracking and mapping for small AR workspaces". In: *2007 6th IEEE and ACM international symposium on mixed and augmented reality*. IEEE. 2007, pp. 225–234.

[228] Tomasz Kocejko, Adam Bujnowski, and Jerzy Wtorek. "Eye-mouse for disabled". In: *Human-computer systems interaction*. Springer, 2009, pp. 109–122.

[229] Sebastian Kohn et al. "Towards a Real-Time Environment Reconstruction for VR-Based Teleoperation Through Model Segmentation". In: *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2018, pp. 1–9.

[230] Timo Koskela et al. "AVATAREX: Telexistence System based on Virtual Avatars". In: *Proceedings of the 9th Augmented Human International Conference*. ACM. 2018, p. 13.

[231] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. "Imagenet classification with deep convolutional neural networks". In: *Communications of the ACM* 60.6 (2017), pp. 84–90.

[232] Norbert Krüger, Michael Pötzsch, and Christoph von der Malsburg. "Determination of face position and pose with a learned representation based on labelled graphs". In: *Image and vision computing* 15.8 (1997), pp. 665–673.

[233] Dennis Krupke et al. "Comparison of Multimodal Heading and Pointing Gestures for Co-Located Mixed Reality Human-Robot Interaction". In: *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2018, pp. 1–9.

[234] Kota Kumagai, Satoshi Hasegawa, and Yoshio Hayasaki. "Volumetric bubble display". In: *Optica* 4.3 (2017), pp. 298–302.

[235] Archek Praveen Kumar et al. "Continuous Telugu Speech Recognition through Combined Feature Extraction by MFCC and DWPD Using HMM based DNN Techniques". In: *International Journal of Pure and Applied Mathematics* 114.11 (2017), pp. 187–197.

[236] DJ Kupetz, SA Wentzell, and BF BuSha. "Head motion controlled power wheelchair". In: *Proceedings of the 2010 IEEE 36th Annual Northeast Bioengineering Conference (NEBEC)*. IEEE. 2010, pp. 1–2.

[237] Junghyun Kwon and Frank C Park. "Natural movement generation using hidden markov models and principal components". In: *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* 38.5 (2008), pp. 1184–1194.

[238] Ondrej Kyjaneka et al. "Implementation of an Augmented Reality AR workflow for Human Robot Collaboration in Timber Prefabrication". In: *ISARC. Proceedings of the International Symposium on Automation and Robotics in Construction*. Vol. 36. IAARC Publications. 2019, pp. 1223–1230.

[239] Jens Lambrecht and Jörg Krüger. "Spatial programming for industrial robots based on gestures and augmented reality". In: *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE. 2012, pp. 466–472.

[240] Jens Lambrecht, Hendrik Walzel, and Jörg Krüger. "Robust finger gesture recognition on handheld devices for spatial programming of industrial robots". In: *RO-MAN, 2013 IEEE*. IEEE. 2013, pp. 99–106.

[241] Leslie Lamport. *Latex*. Addison-Wesley, 1994.

[242] Belinda Lange et al. "Development and evaluation of low cost game-based balance rehabilitation tool using the Microsoft Kinect sensor". In: *2011 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*. IEEE. 2011, pp. 1831–1834.

[243] Knut Langhans et al. "FELIX 3D display: Human-machine interface for interactive real three-dimensional imaging". In: *International Conference on Virtual Storytelling*. Springer. 2005, pp. 22–31.

[244] Tobias Langlotz et al. "Robust detection and tracking of annotations for outdoor augmented reality browsing". In: *Computers & graphics* 35 (Aug. 2011), pp. 831–840. DOI: 10.1016/j.cag.2011.04.004.

[245] Heiner Lasi et al. "Industry 4.0". In: *Business & information systems engineering* 6.4 (2014), pp. 239–242.

[246] Harry T Lawless et al. "Metallic taste from electrical and chemical stimulation". In: *Chemical senses* 30.3 (2005), pp. 185–194.

[247] Viet Bac Le, Laurent Besacier, and Tanja Schultz. "Acoustic-phonetic unit similarities for context dependent acoustic model portability". In: *2006 IEEE International Conference on Acoustics Speech and Signal Processing Proceedings*. Vol. 1. IEEE. 2006, pp. I–I.

[248] Jung-Min Lee et al. "Active insp ection supporting system based on mixed reality after design and manufacture in an offshore structure". In: *Journal of mechanical science and technology* 24.1 (Jan. 2010), pp. 197–202. DOI: 10.1007/s12206-009-1129-2.

[249] Claus Lenz and Alois Knoll. "Mechanisms and capabilities for human robot collaboration". In: *The 23rd IEEE International Symposium on Robot and Human Interactive Communication*. IEEE. 2014, pp. 666–671.

[250] Florian Leutert, Christian Herrmann, and Klaus Schilling. "A spatial augmented reality system for intuitive display of robotic data". In: *Proceedings of the 8th ACM/IEEE international conference on Human-robot interaction*. IEEE Press. 2013, pp. 179–180.

[251] Hsuan Lin, Yu-Chen Hsieh, and Wei Lin. "A Preliminary Study on How the Icon Composition and Background of Graphical Icons Affect Users' Preference Levels". In: *International Conference on Human Aspects of IT for the Aged Population*. Springer. 2016, pp. 360–370.

[252] Hsuan Lin, Yu-Chen Hsieh, and Fong-Gong Wu. "A study on the relationships between different presentation modes of graphical icons and users' attention". In: *Computers in Human Behavior* 63 (2016), pp. 218–228.

[253] Hsuan Lin et al. "How different presentation modes of graphical icons affect viewers' first fixation and attention". In: *International Conference on Universal Access in Human-Computer Interaction*. Springer. 2015, pp. 226–237.

[254] Junguo Lin et al. "Retinal projection head-mounted display". In: *Frontiers of Optoelectronics* 10.1 (2017), pp. 1–8.

[255] Irma Lindt et al. “Combining multiple gaming interfaces in epidemic menace”. In: *CHI'06 Extended Abstracts on Human Factors in Computing Systems*. ACM. 2006, pp. 213–218.

[256] Amy Linklater and Jeff Slutz. “Exploring the large amplitude multi-mode aerospace research simulator's motion drive algorithms”. In: *AIAA Modeling and Simulation Technologies Conference and Exhibit*. 2007, p. 6470.

[257] Christian Linn et al. “Virtual remote inspection—A new concept for virtual reality enhanced real-time maintenance”. In: *2017 23rd International Conference on Virtual System & Multimedia (VSMM)*. Dublin, Ireland. IEEE. 2017, pp. 1–6.

[258] Jeffrey I Lipton, Aidan J Fay, and Daniela Rus. “Baxter's homunculus: Virtual reality spaces for teleoperation in manufacturing”. In: *IEEE Robotics and Automation Letters* 3.1 (2017), pp. 179–186.

[259] Diyi Liu, Jun Kinugawa, and Kazuhiro Kosuge. “A projection-based making-human-feel-safe system for human-robot cooperation”. In: *2016 IEEE International Conference on Mechatronics and Automation*. IEEE. 2016, pp. 1101–1106.

[260] Hangxin Liu et al. “Interactive robot knowledge patching using augmented reality”. In: *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2018, pp. 1947–1954.

[261] Hongyi Liu and Lihui Wang. “An AR-based worker support system for human-robot collaboration”. In: *Procedia Manufacturing* 11 (2017), pp. 22–30.

[262] Kun Liu et al. “Attention recognition of drivers based on head pose estimation”. In: *2008 IEEE Vehicle Power and Propulsion Conference*. IEEE. 2008, pp. 1–5.

[263] Y. Liu and Y. Zhang. “Toward Welding Robot With Human Knowledge: A Remotely-Controlled Approach”. In: *IEEE Transactions on Automation Science and Engineering* 12.2 (2015), pp. 769–774.

[264] Wai Han Lo and Ka Lun Benjamin Cheng. “Does virtual reality attract visitors? The mediating effect of presence on consumer response in virtual reality tourism advertising”. In: *Information Technology & Tourism* 22.4 (2020), pp. 537–562.

[265] Jacobus C Lock et al. “Bone-Conduction Audio Interface to Guide People with Visual Impairments”. In: *International Conference on Smart City and Informatization*. Springer. 2019, pp. 542–553.

[266] Carl Eugene Loeffler. “Distributed virtual reality: Applications for education, entertainment and industry”. In: *Telektronikk* 89 (1993), pp. 83–83.

[267] Cephise Louison et al. "Operators' accessibility studies for assembly and maintenance scenarios using virtual reality". In: *Fusion Engineering and Design* 124 (2017), pp. 610–614.

[268] David G Lowe. "Distinctive image features from scale-invariant keypoints". In: *International journal of computer vision* 60.2 (2004), pp. 91–110.

[269] Qingshan Luo et al. "Human action detection via boosted local motion histograms". In: *Machine Vision and Applications* 21.3 (2010), pp. 377–389.

[270] Justin A MacDonald, Paula P Henry, and Tomasz R Letowski. "Spatial audio through a bone conduction interface: Audición espacial a través de una interfase de conducción ósea". In: *International journal of audiology* 45.10 (2006), pp. 595–599.

[271] Michael Macedonia et al. "NPSNET: a multi-player 3D virtual environment over the Internet". In: Jan. 1995, pp. 93–94, 210. DOI: 10.1145/199404.199419.

[272] I Scott MacKenzie and Yves Guiard. "The two-handed desktop interface: Are we there yet?" In: *CHI'01 extended abstracts on Human factors in computing systems*. 2001, pp. 351–352.

[273] James MacQueen et al. "Some methods for classification and analysis of multivariate observations". In: *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*. Vol. 1. 14. Oakland, CA, USA. 1967, pp. 281–297.

[274] Sotiris Makris et al. "Augmented reality system for operator support in human–robot collaborative assembly". In: *CIRP Annals-Manufacturing Technology* 65.1 (2016), pp. 61–64.

[275] Alessio Malizia and Andrea Bellucci. "The artificiality of natural user interfaces". In: *Communications of the ACM* 55.3 (2012), pp. 36–38.

[276] Ivo Maly, David Sedlacek, and Paulo Leitao. "Augmented reality experiments with industrial robot in industry 4.0 environment". In: *2016 IEEE 14th International Conference on Industrial Informatics (INDIN)*. IEEE. 2016, pp. 176–181.

[277] Levi Manring et al. "Augmented Reality for Interactive Robot Control". In: *Special Topics in Structural Dynamics & Experimental Techniques, Volume 5*. Springer, 2020, pp. 11–18.

[278] Federico Manuri, Francesco De Pace, and Andrea Sanna. *Augmented Reality for Human-Robot Interaction in Industry*. 2019.

[279] Federico Manuri et al. "A workflow analysis for implementing AR-based maintenance procedures". In: *Proc. AVR 2014*. Lecce: Springer, 2014, pp. 185–200.

[280] Ana I Maqueda et al. "Human–computer interaction based on visual hand-gesture recognition using volumetric spatiograms of local binary patterns". In: *Computer Vision and Image Understanding* 141 (2015), pp. 126–137.

[281] André Martin. "Cathode ray tubes for industrial and military applications". In: *Advances in electronics and electron physics* 67 (1986), pp. 183–328.

[282] Andres Martin-Barrio et al. "Application of immersive technologies and natural language to hyper-redundant robot teleoperation". In: *Virtual Reality* (2019), pp. 1–15.

[283] Carlos Mateo et al. "Hammer: An Android based application for end-user industrial robot programming". In: *Proc. MESA*. Senigallia: IEEE, 2014, pp. 1–6.

[284] Zdeněk Materna et al. "Interactive Spatial Augmented Reality in Collaborative Robot Programming: User Experience Evaluation". In: *2018 27th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*. IEEE. 2018, pp. 80–87.

[285] Zdeněk Materna et al. "Using persona, scenario, and use case to develop a human-robot augmented reality collaborative workspace". In: *Proceedings of the Companion of the 2017 ACM/IEEE International Conference on Human-Robot Interaction*. ACM. 2017, pp. 201–202.

[286] Dan Maynes-Aminzade. "Edible bits: Seamless interfaces between people, data and food". In: *Conference on Human Factors in Computing Systems (CHI'05)-Extended Abstracts*. Citeseer. 2005, pp. 2207–2210.

[287] Jesus Mayor, Laura Raya, and Alberto Sanchez. "A comparative study of virtual reality methods of interaction and locomotion based on presence, cybersickness and usability". In: *IEEE Transactions on Emerging Topics in Computing* (2019).

[288] Michael McCauley and Thomas Sharkey. "Cybersickness: Perception of Self-Motion in Virtual Environment." In: *Presence* 1 (Jan. 1992), pp. 311–318. DOI: 10.1162/pres.1992.1.3.311.

[289] Frank McCown. *History of the Graphical User Interface (GUI)*. 2016.

[290] Stephen J McKenna and Shaogang Gong. "Real-time face pose estimation". In: *Real-Time Imaging* 4.5 (1998), pp. 333–347.

[291] Margaret McLaughlin et al. "Integrated voice and haptic support for telerehabilitation". In: *Fourth Annual IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOMW'06)*. IEEE. 2006, 4–pp.

[292] Courtney McNamara, Matthew Proetsch, and Nelson Lerma. "Investigating low-cost virtual reality technologies in the context of an immersive maintenance training application". In: *International Conference on Virtual, Augmented and Mixed Reality*. Toronto, Canada. Springer. 2016, pp. 621–632.

[293] George Michalos et al. "Augmented reality (AR) applications for supporting human-robot interactive cooperation". In: *Procedia CIRP* 41 (2016), pp. 370–375.

[294] George Michalos et al. "Seamless human robot collaborative assembly–An automotive case study". In: *Mechatronics* 55 (2018), pp. 194–211.

[295] Paul Milgram and Fumio Kishino. "A taxonomy of mixed reality visual displays". In: *IEICE TRANSACTIONS on Information and Systems* 77.12 (1994), pp. 1321–1329.

[296] M Moreno et al. "Realtime local navigation for the blind: detection of lateral doors and sound interface". In: *Procedia Computer Science* 14 (2012), pp. 74–82.

[297] Christian Moro, Zane Štromberga, and Allan Stirling. "Virtualisation devices for student learning: Comparison between desktop-based (Oculus Rift) and mobile-based (Gear VR) virtual reality in medical and health science education". In: *Australasian Journal of Educational Technology* 33.6 (2017).

[298] Annette Mossel et al. "ARTiFICe - Augmented Reality Framework for Distributed Collaboration". In: *The International Journal of Virtual Reality* 11 (Jan. 2012), pp. 1–7. DOI: 10.20870/IJVR.2012.11.3.2845.

[299] D Mourtzis, V Zogopoulos, and E Vlachou. "Augmented reality application to support remote maintenance as a service in the robotics industry". In: *Procedia CIRP* 63 (2017), pp. 46–51.

[300] Muhanna A Muhanna. "Virtual reality and the CAVE: Taxonomy, interaction challenges and research directions". In: *Journal of King Saud University-Computer and Information Sciences* 27.3 (2015), pp. 344–361.

[301] Martin Murer, Ilhan Aslan, and Manfred Tscheligi. "LOLL io: exploring taste as playful modality". In: *Proceedings of the 7th International Conference on Tangible, Embedded and Embodied Interaction*. 2013, pp. 299–302.

[302] Erik Murphy-Chutorian and Mohan Manubhai Trivedi. "Head pose estimation in computer vision: A survey". In: *IEEE transactions on pattern analysis and machine intelligence* 31.4 (2008), pp. 607–626.

[303] Brad A Myers et al. "Extending the windows desktop interface with connected handheld computers". In: *WSS'00 Proceedings of the 4th Conference on USENIX Windows Systems Symposium*. Vol. 4. 2000, p. 10.

[304] Mahdi Nabiyouni et al. “Comparing the performance of natural, semi-natural, and non-natural locomotion techniques in virtual reality”. In: *2015 IEEE Symposium on 3D User Interfaces (3DUI)*. IEEE. 2015, pp. 3–10.

[305] Abdeldjallil Naceri et al. “Towards a Virtual Reality Interface for Remote Robotic Teleoperation”. In: *2019 19th International Conference on Advanced Robotics (ICAR)*. IEEE. 2019, pp. 284–289.

[306] Leonid Naimark and Eric Foxlin. “Circular data matrix fiducial system and robust image processing for a wearable vision-inertial self-tracker”. In: *Proceedings. International Symposium on Mixed and Augmented Reality*. IEEE. 2002, pp. 27–36.

[307] Nassir Navab, Sandro-Michael Heining, and Joerg Traub. “Camera augmented mobile C-arm (CAMC): calibration, accuracy study, and clinical applications”. In: *IEEE transactions on medical imaging* 29.7 (2009), pp. 1412–1423.

[308] Andrew YC Nee et al. “Augmented reality applications in design and manufacturing”. In: *CIRP annals* 61.2 (2012), pp. 657–679.

[309] Thomas Nescher, Ying-Yin Huang, and Andreas Kunz. “Planning redirection techniques for optimal free walking experience using model predictive control”. In: *2014 IEEE Symposium on 3D User Interfaces (3DUI)*. IEEE. 2014, pp. 111–118.

[310] GB Newby. “Virtual reality and the entertainment industry”. In: *Bulletin of the American Society for Information Science* 21.1 (1994), pp. 20–21.

[311] Richard A Newcombe et al. “Kinectfusion: Real-time dense surface mapping and tracking”. In: *2011 10th IEEE international symposium on mixed and augmented reality*. IEEE. 2011, pp. 127–136.

[312] David Nistér. “An efficient solution to the five-point relative pose problem”. In: *IEEE transactions on pattern analysis and machine intelligence* 26.6 (2004), pp. 756–770.

[313] S Nivetha. “A Survey on Speech Feature Extraction and Classification Techniques”. In: *2020 International Conference on Inventive Computation Technologies (ICICT)*. IEEE. 2020, pp. 48–53.

[314] Sourabh Niyogi and William T Freeman. “Example-based head tracking”. In: *Proceedings of the second international conference on automatic face and gesture recognition*. IEEE. 1996, pp. 374–378.

[315] Takuya Nojima et al. “Designing augmented sports: Merging physical sports and virtual world game concept”. In: *International Conference on Human Interface and the Management of Information*. Springer. 2018, pp. 403–414.

[316] Marianna Obrist et al. "Temporal, affective, and embodied characteristics of taste experiences: a framework for design". In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 2014, pp. 2853–2862.

[317] Yoichi Ochiai et al. "Fairy lights in femtoseconds: aerial and volumetric graphics rendered by focused femtosecond laser combined with computational holographic fields". In: *ACM Transactions on Graphics (TOG)* 35.2 (2016), pp. 1–14.

[318] Lael U Odhner, Raymond R Ma, and Aaron M Dollar. "Open-loop precision grasping with underactuated hands inspired by a human manipulation strategy". In: *IEEE Transactions on Automation Science and Engineering* 10.3 (2013), pp. 625–633.

[319] Jan Ohlenburg et al. "The MORGAN framework: enabling dynamic multi-user AR and VR projects". In: Jan. 2004, pp. 166–169. DOI: 10.1145/1077534.1077568.

[320] Takehiko Ohno. "One-point calibration gaze tracking method". In: *Proceedings of the 2006 symposium on Eye tracking research & applications*. 2006, pp. 34–34.

[321] SK Ong, JWS Chong, and AYC Nee. "A novel AR-based robot programming and path planning methodology". In: *Robotics and Computer-Integrated Manufacturing* 26.3 (2010), pp. 240–249.

[322] SK Ong, ML Yuan, and AYC Nee. "Augmented reality applications in manufacturing: a survey". In: *International journal of production research* 46.10 (2008), pp. 2707–2742.

[323] SK Ong et al. "Augmented reality-assisted robot programming system for industrial applications". In: *Robotics and Computer-Integrated Manufacturing* 61 (2020), pp. 1–7.

[324] Soh-Khim Ong, JWS Chong, and Andrew YC Nee. "Methodologies for immersive robot programming in an augmented reality environment". In: *Proceedings of the 4th international conference on computer graphics and interactive techniques in Australasia and Southeast Asia*. ACM. 2006, pp. 237–244.

[325] G Ortega et al. "Usefulness of a head mounted monitor device for viewing intraoperative fluoroscopy during orthopaedic procedures". In: *Archives of orthopaedic and trauma surgery* 128.10 (2008), pp. 1123–1126.

[326] M Ostanin et al. "Multi robots interactive control using mixed reality". In: *International Journal of Production Research* (2020), pp. 1–13.

[327] Nobuyuki Otsu. "A threshold selection method from gray-level histograms". In: *IEEE transactions on systems, man, and cybernetics* 9.1 (1979), pp. 62–66.

[328] John O Oyekan et al. “The effectiveness of virtual environments in developing collaborative strategies between industrial robots and humans”. In: *Robotics and Computer-Integrated Manufacturing* 55 (2019), pp. 41–54.

[329] GF Page. *MULTIPLE VIEW GEOMETRY IN COMPUTER VISION, by Richard Hartley and Andrew Zisserman, CUP, Cambridge, UK, 2003, vi+ 560 pp., ISBN 0-521-54051-8.(Paperback£ 44.95)*. 2005.

[330] Yun Suen Pai, Hwa Jen Yap, and Ramesh Singh. “Augmented reality–based programming, planning and simulation of a robotic work cell”. In: *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture* 229.6 (2015), pp. 1029–1045.

[331] Federica Pallavicini et al. “What distinguishes a traditional gaming experience from one in virtual reality? An exploratory study”. In: *International Conference on Applied Human Factors and Ergonomics*. Springer. 2017, pp. 225–231.

[332] Riccardo Palmarini et al. “Designing an AR interface to improve trust in human-robots collaboration”. In: *Procedia CIRP* 70.1 (2018), pp. 350–355.

[333] Chris Panou et al. “An architecture for mobile outdoors augmented reality for cultural heritage”. In: *ISPRS International Journal of Geo-Information* 7.12 (2018), p. 463.

[334] Gabriel Pantoja, Luis Eduardo Garza, and Eduardo Gonzalez Mendivil. “Augmented reality in pneumatic conveying system: fuller pump dry material line charger”. In: *Proc. CISTI*. Barcelona: IEEE, 2014, pp. 1–5.

[335] Stergios Papanastasiou et al. “Towards seamless human robot collaboration: integrating multimodal interaction”. In: *The International Journal of Advanced Manufacturing Technology* 105.9 (2019), pp. 3881–3897.

[336] Biswaksen Patnaik, Andrea Batch, and Niklas Elmqvist. “Information olfactation: Harnessing scent to convey data”. In: *IEEE transactions on visualization and computer graphics* 25.1 (2018), pp. 726–736.

[337] Huaishu Peng et al. “RoMA: Interactive fabrication with augmented reality and a robotic 3D printer”. In: *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. ACM. 2018, p. 579.

[338] Katharina Pentenrieder et al. “Augmented Reality-based factory planning-an application tailored to industrial needs”. In: *2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality*. IEEE. 2007, pp. 31–42.

[339] Luis Pérez et al. “Industrial robot control and operator training using virtual reality interfaces”. In: *Computers in Industry* 109 (2019), pp. 114–120.

[340] R Perez-Ubeda et al. "Study of the application of a collaborative robot for machining tasks". In: *Procedia Manufacturing* 41 (2019), pp. 867–874.

[341] Grégory Petit et al. "Refreshable tactile graphics applied to schoolbook illustrations for students with visual impairment". In: *Proceedings of the 10th international ACM SIGACCESS conference on Computers and accessibility*. 2008, pp. 89–96.

[342] Anna Petrovskaya and Oussama Khatib. "Global localization of objects via touch". In: *IEEE Transactions on Robotics* 27.3 (2011), pp. 569–585.

[343] Gert Pfurtscheller et al. "Current trends in Graz brain-computer interface (BCI) research". In: *IEEE transactions on rehabilitation engineering* 8.2 (2000), pp. 216–219.

[344] Michele Pirovano. "Kinfu–an open source implementation of Kinect Fusion+ case study: implementing a 3D scanner with PCL". In: *Project Assignment* (2012).

[345] K-H Plattig and J Innitzer. "Taste qualities elicited by electric stimulation of single human tongue papillae". In: *Pflügers Archiv* 361.2 (1976), pp. 115–120.

[346] Jarkko Polvi et al. "Handheld guides in inspection tasks: Augmented reality versus picture". In: *IEEE transactions on visualization and computer graphics* 24.7 (2017), pp. 2118–2128.

[347] Jarkko Polvi et al. *User Interface Design of a SLAM-based Handheld Augmented Reality Work Support System*. Tech. rep. VRSJ Research Report, 2013.

[348] Michal Ponder et al. "VHD++ Development Framework: Towards Extendible, Component Based VR/AR Simulation Engine Featuring Advanced Virtual Character Technologies." In: vol. 2003. Jan. 2003, pp. 96–104. DOI: 10.1109/CGI.2003.1214453.

[349] Chomtip Pornpanomchai et al. "Ad-Smell: Advertising movie with a simple olfactory display". In: *Proceedings of the First International Conference on Internet Multimedia Computing and Service*. 2009, pp. 113–118.

[350] Jess Porter et al. "Mechanisms of scent-tracking in humans". In: *Nature neuroscience* 10.1 (2007), pp. 27–29.

[351] Albert B Pratt. *Weapon*. US Patent 1,183,492. May 1916.

[352] Camilo Perez Quintero et al. "Robot programming through augmented trajectories in augmented reality". In: *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2018, pp. 1838–1844.

[353] Kjetil Raaen and Ivar Kjellmo. "Measuring Latency in Virtual Reality Systems". In: Sept. 2015, pp. 457–462. ISBN: 978-3-319-24588-1.

[354] Lawrence R Rabiner. "A tutorial on hidden Markov models and selected applications in speech recognition". In: *Proceedings of the IEEE* 77.2 (1989), pp. 257–286.

[355] Amer Al-Rahayfeh and Miad Faezipour. "Eye tracking and head movement detection: A state-of-art survey". In: *IEEE journal of translational engineering in health and medicine* 1 (2013), pp. 2100212–2100212.

[356] Rabie A Ramadan and Athanasios V Vasilakos. "Brain computer interface: control signals review". In: *Neurocomputing* 223 (2017), pp. 26–44.

[357] Perla Ramakrishna et al. "An ar inspection framework: Feasibility study with multiple ar devices". In: *Proc. ISMAR-Adjunct*. Merida: IEEE, 2016, pp. 221–226.

[358] Shubhankar Ranade et al. "Clash tanks: An investigation of virtual and augmented reality gaming experience". In: *2017 Tenth International Conference on Mobile Computing and Ubiquitous Network (ICMU)*. IEEE. 2017, pp. 1–6.

[359] Nimesha Ranasinghe et al. "Tongue mounted interface for digitally actuating the sense of taste". In: *2012 16th International Symposium on Wearable Computers*. IEEE. 2012, pp. 80–87.

[360] Vidas Raudonis, Rimvydas Simutis, and Gintautas Narvydas. "Discrete eye tracking for medical applications". In: *2009 2nd International Symposium on Applied Sciences in Biomedical and Communication Technologies*. IEEE. 2009, pp. 1–6.

[361] Siddharth S Rautaray and Anupam Agrawal. "Vision based hand gesture recognition for human computer interaction: a survey". In: *Artificial intelligence review* 43.1 (2015), pp. 1–54.

[362] Matthias Rauterberg. "An empirical comparison of menu-selection (((CUI) and desktop (GUI) computer programs carried out by beginners and experts". In: *Behaviour & Information Technology* 11.4 (1992), pp. 227–236.

[363] Frédéric Rayar, David Boas, and Rémi Patrizio. "ART-chess: a tangible augmented reality chess on tabletop". In: *Proceedings of the 2015 International Conference on Interactive Tabletops & Surfaces*. 2015, pp. 229–233.

[364] Torsten Reiners, Lincoln C Wood, and Sue Gregory. "Experimental study on consumer-technology supported authentic immersion in virtual environments for education and vocational training". In: *Rhetoric and Reality: Critical perspectives on educational technology. Proceedings of ascilite Dunedin 2014* (2014).

[365] Gunther Reinhart, Wolfgang Vogl, and Ingo Kresse. "A projection-based user interface for industrial robots". In: *2007 IEEE Symposium on Virtual Environments, Human-Computer Interfaces and Measurement Systems*. IEEE. 2007, pp. 67–71.

[366] Donghao Ren et al. "Evaluating wide-field-of-view augmented reality with mixed reality simulation". In: *2016 IEEE Virtual Reality (VR)*. IEEE. 2016, pp. 93–102.

[367] Matthew R Rhea. "Determining the magnitude of treatment effects in strength training research through the use of the effect size". In: *Journal of strength and conditioning research* 18 (2004), pp. 918–920.

[368] Ciarán Robinson. *Game Audio with FMOD and Unity*. Routledge, 2019.

[369] Damien Rompapas et al. "HoloRoyale: A Large Scale High Fidelity Augmented Reality Game". In: *The 31st Annual ACM Symposium on User Interface Software and Technology Adjunct Proceedings*. ACM. 2018, pp. 163–165.

[370] Robert Rosenthal, Harris Cooper, L Hedges, et al. "Parametric measures of effect size". In: *The handbook of research synthesis* 621.2 (1994), pp. 231–244.

[371] Edward Rosten and Tom Drummond. "Machine learning for high-speed corner detection". In: *European conference on computer vision*. Springer. 2006, pp. 430–443.

[372] Ernesto de la Rubia and Antonio Diaz-Estrella. "Natural locomotion based on foot-mounted inertial sensors in a wireless virtual reality system". In: *Presence* 24.4 (2015), pp. 298–321.

[373] Emanuele Ruffaldi et al. "Third point of view augmented reality for robot intentions visualization". In: *Proc. AVR 2016*. Otranto: Springer, 2016, pp. 471–478.

[374] Emanuele Ruffaldi et al. "Third point of view augmented reality for robot intentions visualization". In: *International Conference on Augmented Reality, Virtual Reality and Computer Graphics*. Springer. 2016, pp. 471–478.

[375] Ahmed Sajjad et al. "Speaker Identification & Verification Using MFCC & SVM". In: *International Research Journal of Engineering and Technology (IRJET)* 4.02 (2017).

[376] Ali Samini and Karljohan Lundin Palmerius. "A study on improving close and distant device movement pose manipulation for hand-held augmented reality". In: *Proceedings of the 22nd ACM Conference on Virtual Reality Software and Technology*. 2016, pp. 121–128.

[377] Andrea Sanna et al. “A flexible AR-based training system for industrial maintenance”. In: *Proc. AVR 2015*. Lecce: Springer, 2015, pp. 314–331.

[378] Arpita Ray Sarkar, G Sanyal, and SJIJOCA Majumder. “Hand gesture recognition systems: a survey”. In: *International Journal of Computer Applications* 71.15 (2013).

[379] Akira Sasou. “Acoustic head orientation estimation applied to powered wheelchair control”. In: *2009 Second International Conference on Robot Communication and Coordination*. IEEE. 2009, pp. 1–6.

[380] Paula Savioja et al. “Developing a mobile, service-based augmented reality tool for modern maintenance work”. In: *International Conference on Virtual Reality*. Springer. 2007, pp. 554–563.

[381] Lina Sawalha et al. “A large 3D swept-volume video display”. In: *Journal of Display Technology* 8.5 (2012), pp. 256–268.

[382] Shreyashi Narayan Sawant and MS Kumbhar. “Real time sign language recognition using pca”. In: *2014 IEEE International Conference on Advanced Communications, Control and Computing Technologies*. IEEE. 2014, pp. 1412–1415.

[383] Gerhard Schall et al. “Handheld augmented reality for underground infrastructure visualization”. In: *Personal and ubiquitous computing* 13.4 (2009), pp. 281–291.

[384] Thomas Schlömer et al. “Gesture recognition with a Wii controller”. In: *Proceedings of the 2nd international conference on Tangible and embedded interaction*. 2008, pp. 11–14.

[385] Dieter Schmalstieg. “Augmented reality techniques in games”. In: *Fourth IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR’05)*. IEEE. 2005, pp. 176–177.

[386] Dieter Schmalstieg and Tobias Hollerer. *Augmented reality: principles and practice*. Addison-Wesley Professional, 2016.

[387] Dieter Schmalstieg et al. “Studierstube-an environment for collaboration in augmented reality”. In: *CVE’96 Workshop Proceedings*. Vol. 19. 1996.

[388] Dominik Schmidt et al. “Level-ups: Motorized stilts that simulate stair steps in virtual reality”. In: *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. 2015, pp. 2157–2160.

[389] Jürgen P. Schulze et al. “CalVR: an advanced open source virtual reality software framework”. In: *The Engineering Reality of Virtual Reality 2013*. Ed. by Margaret Dolinsky and Ian E. McDowall. Vol. 8649. International Society for Optics and Photonics. SPIE, 2013, pp. 1–8. DOI: 10.1117/12.2005241. URL: https://doi.org/10.1117/12.2005241.

[390] Bernd Schwald et al. “STARMATE: Using Augmented Reality technology for computer guided maintenance of complex mechanical elements”. In: *E-work and ECommerce* 1 (2001), pp. 196–202.

[391] Scott et al. “Multi-User Framework for Collaboration and Co-Creation in Virtual Reality”. In: (June 2017).

[392] Edgar Seemann, Kai Nickel, and Rainer Stiefelhagen. “Head pose estimation using stereo vision for human-robot interaction”. In: *Sixth IEEE International Conference on Automatic Face and Gesture Recognition, 2004. Proceedings.* IEEE. 2004, pp. 626–631.

[393] Javier Servan et al. “Assembly work instruction deployment using augmented reality”. In: *Key Engineering Materials.* Vol. 502. Trans Tech Publ. 2012, pp. 25–30.

[394] Shaham Shabani and Yaser Norouzi. “Speech recognition using Principal Components Analysis and Neural Networks”. In: *2016 IEEE 8th International Conference on Intelligent Systems (IS).* IEEE. 2016, pp. 90–95.

[395] Anjana Sharma and Pawanesh Abrol. “Eye gaze techniques for human computer interaction: A research survey”. In: *International Journal of Computer Applications* 71.9 (2013).

[396] Mark J Shensa. “The discrete wavelet transform: wedding the a trous and Mallat algorithms”. In: *IEEE Transactions on signal processing* 40.10 (1992), pp. 2464–2482.

[397] William R Sherman and Alan B Craig. *Understanding virtual reality: Interface, application, and design.* Morgan Kaufmann, 2018.

[398] Lei Shi et al. “A Performance Analysis of Invariant Feature Descriptors in Eye Tracking based Human Robot Collaboration”. In: *2019 5th International Conference on Control, Automation and Robotics (ICCAR).* IEEE. 2019, pp. 256–260.

[399] Takashi Shibata. “Head mounted display”. In: *Displays* 23.1-2 (2002), pp. 57–64.

[400] A. Simonič. “A Construction of Lomonosov Functions and Applications to the Invariant Subspace Problem”. In: *Pacific J. Math.* 175 (1996), pp. 257–270.

[401] A. Simonič. “An Extension of Lomonosov’s Techniques to Non-Compact Operators”. PhD thesis. Dalhousie University, Department of Mathematics, Statistics, & Computing Science, 1994.

[402] A. Simonič. “Grupe Operatorjev s Pozitivnim Spektrom”. MA thesis. Univerza v Ljubljani, FNT, Oddelek za Matematiko, 1990.

[403] A. Simonič. "Matrix Groups with Positive Spectra". In: *Linear Algebra Appl.* 173 (1992), pp. 57–76.

[404] A. Simonič. "Notes on Subharmonic Functions". Lecture Notes, Dalhousie University, Department of Mathematics, Statistics, & Computing Science. 1991.

[405] Elliot Singer et al. "Acoustic, phonetic, and discriminative approaches to automatic language identification". In: *Eighth European conference on speech communication and technology*. 2003.

[406] Virk Damanbir Singh and VK Banga. "Overloading failures in robot manipulators". In: *International Conference on Trends in Electrical, Electronics and Power Engineering (ICTEEP'2012)/Planetary Scientific Research Centre*. 2012, pp. 15–16.

[407] Enrico Sita et al. "Towards multimodal interactions: robot jogging in mixed reality". In: *Proceedings of the 23rd ACM Symposium on Virtual Reality Software and Technology*. ACM. 2017, pp. 1–2.

[408] Ranganatha Sitaram et al. "fMRI brain-computer interfaces". In: *IEEE Signal processing magazine* 25.1 (2007), pp. 95–106.

[409] Nancy A Skopp et al. "A pilot study of the virtusphere as a virtual reality enhancement". In: *International Journal of Human-Computer Interaction* 30.1 (2014), pp. 24–31.

[410] DE Smalley et al. "A photophoretic-trap volumetric display". In: *Nature* 553.7689 (2018), pp. 486–490.

[411] Rajinder S Sodhi et al. "BeThere: 3D mobile collaboration with spatial input". In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM. 2013, pp. 179–188.

[412] Hyunyoung Song et al. "PenLight: combining a mobile projector and a digital pen for dynamic visual overlay". In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 2009, pp. 143–152.

[413] Alois Sontacchi, Michael Strauß, and Robert Holdrich. "Audio interface for immersive 3D-audio desktop applications". In: *IEEE International Symposium on Virtual Environments, Human-Computer Interfaces and Measurement Systems, 2003. VECIMS'03. 2003*. IEEE. 2003, pp. 179–182.

[414] Charles Spence et al. "Digitizing the chemical senses: possibilities & pitfalls". In: *International Journal of Human-Computer Studies* 107 (2017), pp. 62–74.

[415] Sujith Srinivasan and Kim L Boyer. "Head pose estimation using view based eigenspaces". In: *Object recognition supported by user interaction for service robots*. Vol. 4. IEEE. 2002, pp. 302–305.

[416] Darko Stanimirovic et al. "[Poster] A Mobile Augmented reality system to assist auto mechanics". In: *Proc. ISAMR*. Munich: IEEE, 2014, pp. 305–306.

[417] Christopher Stapleton et al. "Applying mixed reality to entertainment". In: *Computer* 35.12 (2002), pp. 122–124.

[418] Robert J Stone, Peter B Panfilov, and Valentin E Shukshunov. "Evolution of aerospace simulation: From immersive Virtual Reality to serious games". In: *Proceedings of 5th International Conference on Recent Advances in Space Technologies-RAST2011*. IEEE. 2011, pp. 655–662.

[419] Didier Stricker, Gundrun Klinker, and Dirk Reiners. "A fast and robust line-based optical tracker for augmented reality applications". In: *International Workshop on Augmented Reality (IWAR'98)*. 1999, pp. 129–145.

[420] Carolyn Sumners, Patricia Reiff, and Wolfgang Weber. "Learning in an immersive digital theater". In: *Advances in Space Research* 42.11 (2008), pp. 1848–1854.

[421] Da Sun et al. "A new mixed-reality-based teleoperation system for telepresence and maneuverability enhancement". In: *IEEE Transactions on Human-Machine Systems* 50.1 (2020), pp. 55–67.

[422] Hongyu Sun et al. "On-line EEG classification for brain-computer interface based on CSP and SVM". In: *2010 3rd International Congress on Image and Signal Processing*. Vol. 9. IEEE. 2010, pp. 4105–4108.

[423] Qi Sun, Li-Yi Wei, and Arie Kaufman. "Mapping virtual and physical reality". In: *ACM Transactions on Graphics (TOG)* 35.4 (2016), pp. 1–12.

[424] Gyung Tak Sung and Inderbir S Gill. "Robotic laparoscopic surgery: a comparison of the da Vinci and Zeus systems". In: *Urology* 58.6 (2001), pp. 893–898. ISSN: 0090-4295. DOI: https://doi.org/10.1016/S0090-4295(01)01423-6. URL: http://www.sciencedirect.com/science/article/pii/S0090429501014236.

[425] Ivan Sutherland. "The ultimate display". In: (1965).

[426] Ivan E Sutherland. "A head-mounted three dimensional display". In: *Proceedings of the December 9-11, 1968, fall joint computer conference, part I*. 1968, pp. 757–764.

[427] Ivan E Sutherland. "Sketchpad a man-machine graphical communication system". In: *Simulation* 2.5 (1964), R–3.

[428] Satoshi Suzuki et al. "Topological structural analysis of digitized binary images by border following". In: *Computer vision, graphics, and image processing* 30.1 (1985), pp. 32–46.

[429] Zsolt Szalavári, Erik Eckstein, and Michael Gervautz. "Collaborative gaming in augmented reality". In: *VRST*. Vol. 98. 1998, pp. 195–204.

[430] Matthew Tait and Mark Billinghurst. "[Poster] View independence in remote collaboration using AR". In: *Mixed and Augmented Reality (ISMAR), 2014 IEEE International Symposium on*. IEEE. 2014, pp. 309–310.

[431] Matthew Tait and Mark Billinghurst. "The effect of view independence in a collaborative ar system". In: *Computer Supported Cooperative Work (CSCW)* 24.6 (2015), pp. 563–589.

[432] Arthur Tang et al. "Comparative effectiveness of augmented reality in object assembly". In: *Proceedings of the SIGCHI conference on Human factors in computing systems*. 2003, pp. 73–80.

[433] Jianxiong Tang and Jianxin Zhang. "Eye tracking based on grey prediction". In: *2009 First International Workshop on Education Technology and Computer Science*. Vol. 2. IEEE. 2009, pp. 861–864.

[434] Tomohiro Tanikawa et al. "Integrated view-input ar interaction for virtual object manipulation using tablets and smartphones". In: *Proceedings of the 12th International Conference on Advances in Computer Entertainment Technology*. 2015, pp. 1–8.

[435] Matteo Tanzini et al. "A novel human-machine interface for working machines operation". In: *2013 IEEE RO-MAN*. IEEE. 2013, pp. 744–750.

[436] Mahdi Tavakoli, Jay Carriere, and Ali Torabi. "Robotics, smart wearable technologies, and autonomous intelligent systems for healthcare during the COVID-19 pandemic: An analysis of the state of the art and future vision". In: *Advanced Intelligent Systems* (2020), p. 2000071.

[437] Pedro Tavares et al. "Collaborative Welding System using BIM for Robotic Reprogramming and Spatial Augmented Reality". In: *Automation in Construction* 106 (2019), pp. 1–12.

[438] Franco Tecchia, Leila Alem, and Weidong Huang. "3D helping hands: a gesture based MR system for remote collaboration". In: *Proceedings of the 11th ACM SIGGRAPH International Conference on Virtual-Reality Continuum and its Applications in Industry*. ACM. 2012, pp. 323–328.

[439] Graziano Terenzi and Giuseppe Basile. "Smart maintenance–An augmented reality platform for training and field operations in the manufacturing industry". In: *ARMEDIA Augmented Reality Blog* (2014).

[440] Saravanan Thirumuruganathan. "A detailed introduction to K-nearest neighbor (KNN) algorithm". In: *Retrieved March* 20 (2010), p. 2012.

[441] Bruce Thomas et al. "ARQuake: An outdoor/indoor augmented reality first person application". In: *Digest of Papers. Fourth International Symposium on Wearable Computers*. IEEE. 2000, pp. 139–146.

[442] Bruce Thomas et al. “First person indoor/outdoor augmented reality application: ARQuake”. In: *Personal and Ubiquitous Computing* 6.1 (2002), pp. 75–86.

[443] PC Thomas and WM David. “Augmented reality: An application of heads-up display technology to manual manufacturing processes”. In: *Hawaii international conference on system sciences*. 1992, pp. 659–669.

[444] Wu Ting et al. “EEG feature extraction based on wavelet packet decomposition for brain computer interface”. In: *Measurement* 41.6 (2008), pp. 618–625.

[445] Neha Tiwari et al. “Brain computer interface: A comprehensive survey”. In: *Biologically inspired cognitive architectures* 26 (2018), pp. 118–129.

[446] Andreas Tobergte, Rainer Konietschke, and Gerd Hirzinger. “Planning and control of a teleoperation system for research in minimally invasive robotic surgery”. In: *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2009, pp. 4225–4232.

[447] Mădălina Ioana Toma, Florin Girbacia, and Csaba Antonya. “A comparative evaluation of human interaction for design and assembly of 3D CAD models in desktop and immersive environments”. In: *International Journal on Interactive Design and Manufacturing (IJIDeM)* 6.3 (2012), pp. 179–193.

[448] Maciej Tomczak and Ewa Tomczak. “The need to report effect size estimates revisited. An overview of some recommended measures of effect size”. In: (2014).

[449] Xin Tong et al. “The design of an immersive mobile virtual reality serious game in cardboard head-mounted display for pain management”. In: *International Symposium on Pervasive Computing Paradigms for Mental Health*. Springer. 2015, pp. 284–293.

[450] Vesa Valimaki et al. “Assisted listening using a headset: Enhancing audio perception in real, augmented, and virtual environments”. In: *IEEE Signal Processing Magazine* 32.2 (2015), pp. 92–99.

[451] David B Van de Merwe et al. “Human-Robot Interaction During Virtual Reality Mediated Teleoperation: How Environment Information Affects Spatial Task Performance and Operator Situation Awareness”. In: *International Conference on Human-Computer Interaction*. Springer. 2019, pp. 163–177.

[452] BD Varalakshmi et al. “Haptics: state of the art survey”. In: *International Journal of Computer Science Issues (IJCSI)* 9.5 (2012), p. 234.

[453] Germano Veiga, Pedro Malaca, and Rui Cancela. “Interactive industrial robot programming for the ceramic industry”. In: *International Journal of Advanced Robotic Systems* 10.10 (2013), p. 354.

[454] Elizabeth S Veinott, Barbara G Kanki, and Michael G Shafto. "Identifying human factors issues in aircraft maintenance operations". In: (Jan. 1995).

[455] Predrag Veličković and Miloš Milovanović. "Improvement of the Interaction Model Aimed to Reduce the Negative Effects of Cybersickness in VR Rehab Applications". In: *Sensors* 21.2 (2021), p. 321.

[456] Jonathan Ventura et al. "Evaluating the effects of tracker reliability and field of view on a target following task in augmented reality". In: *Proceedings of the 16th ACM Symposium on Virtual Reality Software and Technology*. ACM. 2009, pp. 151–154.

[457] Lucia Vera et al. "A hybrid virtual-augmented serious game to improve driving safety awareness". In: *International Conference on Advances in Computer Entertainment*. Springer. 2017, pp. 293–310.

[458] Christian Vogel and Norbert Elkmann. "Novel Safety Concept for Safeguarding and Supporting Humans in Human-Robot Shared Workplaces with High-Payload Robots in Industrial Applications". In: *Proceedings of the Companion of the 2017 ACM/IEEE International Conference on Human-Robot Interaction*. ACM. 2017, pp. 315–316.

[459] Christian Vogel, Markus Fritzsche, and Norbert Elkmann. "Safe human-robot cooperation with high-payload robots in industrial applications". In: *2016 11th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*. IEEE. 2016, pp. 529–530.

[460] Christian Vogel, Christoph Walter, and Norbert Elkmann. "A projection-based sensor system for safe physical human-robot collaboration". In: *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*. IEEE. 2013, pp. 5359–5364.

[461] Christian Vogel, Christoph Walter, and Norbert Elkmann. "Exploring the possibilities of supporting robot-assisted work places using a projection-based sensor system". In: *Robotic and Sensors Environments (ROSE), 2012 IEEE International Symposium on*. IEEE. 2012, pp. 67–72.

[462] Christian Vogel, Christoph Walter, and Norbert Elkmann. "Safeguarding and Supporting Future Human-robot Cooperative Manufacturing Processes by a Projection-and Camera-based Technology". In: *Procedia Manufacturing* 11 (2017), pp. 39–46.

[463] Christian Vogel et al. "Towards safe physical human-robot collaboration: A projection-based safety system". In: *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*. IEEE. 2011, pp. 3355–3360.

[464] Harold L Vogel. *Entertainment industry economics: A guide for financial analysis*. Cambridge University Press, 2020.

[465] Peter Vorderer et al. “Mec spatial presence questionnaire”. In: *Retrieved Sept* 18 (2004), p. 2015.

[466] Adam Wagler and Michael D Hanus. “Comparing virtual reality tourism to real-life experience: Effects of presence and engagement on attitude and enjoyment”. In: *Communication Research Reports* 35.5 (2018), pp. 456–464.

[467] Daniel Wagner, Tobias Langlotz, and Dieter Schmalstieg. “Robust and unobtrusive marker tracking on mobile phones”. In: *2008 7th IEEE/ACM International Symposium on Mixed and Augmented Reality*. IEEE. 2008, pp. 121–124.

[468] Yujin Wakita et al. “Information sharing via projection function for coexistence of robot and human”. In: *Autonomous Robots* 10.3 (2001), pp. 267–277.

[469] Dangxiao Wang, Kouhei Ohnishi, and Weiliang Xu. “Multimodal haptic display for virtual reality: A survey”. In: *IEEE Transactions on Industrial Electronics* 67.1 (2019), pp. 610–623.

[470] Junfeng Wang et al. “An augmented reality based system for remote collaborative maintenance instruction of complex products”. In: *Proc. CASE*. Taipei: IEEE, 2014, pp. 309–314.

[471] Xiangyu Wang et al. “Mutual awareness in collaborative design: An Augmented Reality integrated telepresence system”. In: *Computers in Industry* 65.2 (2014), pp. 314–324.

[472] Yijun Wang et al. “BCI competition 2003-data set IV: an algorithm based on CSSD and FDA for classifying single-trial EEG”. In: *IEEE Transactions on Biomedical Engineering* 51.6 (2004), pp. 1081–1086.

[473] Oliver Wasenmüller, Marcel Meyer, and Didier Stricker. “Augmented reality 3d discrepancy check in industrial applications”. In: *Proc. ISMAR*. Merida: IEEE, 2016, pp. 125–134.

[474] J A Waterworth and M H Chignell. *Multimedia Interaction*. 1997.

[475] Sabine Webel, Jens Keil, and Michael Zoellner. “Multi-touch gestural interaction in X3D using hidden Markov models”. In: *Proceedings of the 2008 ACM symposium on Virtual reality software and technology*. 2008, pp. 263–264.

[476] Rong Wen et al. *Intraoperative visual guidance and control interface for augmented reality robotic surgery*. IEEE, 2010.

[477] Rong Wen et al. “Robot-assisted RF ablation with interactive planning and mixed reality guidance”. In: *2012 IEEE/SICE International Symposium on System Integration (SII)*. IEEE. 2012, pp. 31–36.

[478] Thomas Weng et al. “Robot Object Referencing through Legible Situated Projections”. In: *2019 International Conference on Robotics and Automation (ICRA)*. IEEE. 2019, pp. 8004–8010.

[479] David Whitney et al. “Comparing robot grasping teleoperation across desktop and virtual reality with ROS reality”. In: *Robotics Research*. Springer, 2020, pp. 335–350.

[480] David Whitney et al. “Ros reality: A virtual reality framework using consumer-grade hardware for ros-enabled robots”. In: *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2018, pp. 1–9.

[481] Frederic L Wightman and Doris J Kistler. “Headphone simulation of free-field listening. I: stimulus synthesis”. In: *The Journal of the Acoustical Society of America* 85.2 (1989), pp. 858–867.

[482] Andrew D Wilson. “Fast lossless depth image compression”. In: *Proceedings of the 2017 ACM International Conference on Interactive Surfaces and Spaces*. 2017, pp. 100–105.

[483] Graham Wilson and Mark McGill. “Violent video games in virtual reality: Re-evaluating the impact and rating of interactive experiences”. In: *Proceedings of the 2018 Annual Symposium on Computer-Human Interaction in Play*. ACM. 2018, pp. 535–548.

[484] Preston Tunnell Wilson et al. “VR locomotion: walking> walking in place> arm swinging”. In: *Proceedings of the 15th ACM SIGGRAPH Conference on Virtual-Reality Continuum and Its Applications in Industry-Volume 1*. 2016, pp. 243–249.

[485] Josef Wolfartsberger. “Analyzing the potential of Virtual Reality for engineering design review”. In: *Automation in Construction* 104 (2019), pp. 27–37.

[486] R Woodfield. “Virtual reality, videogames and the story of art”. In: (1996).

[487] Heinz Wörn and Joachim Mühling. “Computer-and robot-based operation theatre of the future in cranio-facial surgery”. In: *International congress series*. Vol. 1230. Elsevier. 2001, pp. 753–759.

[488] Xingyu Wu et al. “Point context: an effective shape descriptor for RST-invariant trajectory recognition”. In: *Journal of Mathematical Imaging and Vision* 56.3 (2016), pp. 441–454.

[489] Mengxin Xu, Maria Murcia-Lopez, and Anthony Steed. “Object location memory error in virtual and real environments”. In: *2017 IEEE Virtual Reality (VR)*. IEEE. 2017, pp. 315–316.

[490] Yifei Xu, Jinhua Zeng, and Yaoru Sun. "Head pose recovery using 3D cross model". In: *2012 4th International Conference on Intelligent Human-Machine Systems and Cybernetics*. Vol. 2. IEEE. 2012, pp. 63–66.

[491] Caixia Yang et al. "A gray difference-based pre-processing for gaze tracking". In: *IEEE 10th INTERNATIONAL CONFERENCE ON SIGNAL PROCESSING PROCEEDINGS*. IEEE. 2010, pp. 1293–1296.

[492] Xiaohui Yang et al. "A gaze tracking scheme for eye-based intelligent control". In: *2010 8th World Congress on Intelligent Control and Automation*. IEEE. 2010, pp. 50–55.

[493] Yin Yeqing and Tian Tao. "An new speech recognition method based on prosodic analysis and svm in zhuang language". In: *2011 International Conference on Mechatronic Science, Electric Engineering and Computer (MEC)*. IEEE. 2011, pp. 1209–1212.

[494] Xuyue Yin et al. "VR&AR Combined Manual Operation Instruction System on Industry Products: A Case Study". In: *Virtual Reality and Visualization (ICVRV), 2014 International Conference on*. IEEE. 2014, pp. 65–72.

[495] Mariko Yoshida et al. "Study on Stimulation Effects for Driver Based on Fragrance Presentation." In: *MVA*. 2011, pp. 332–335.

[496] Cik Suhaimi Yusof et al. "Collaborative augmented reality for chess game in handheld devices". In: *2019 IEEE Conference on Graphics and Media (GAME)*. IEEE. 2019, pp. 32–37.

[497] Michael F Zaeh and Wolfgang Vogl. "Interactive laser-projection for programming industrial robots". In: *2006 IEEE/ACM International Symposium on Mixed and Augmented Reality*. IEEE. 2006, pp. 125–128.

[498] Markus Zank and Andreas Kunz. "Using locomotion models for estimating walking targets in immersive virtual environments". In: *2015 International Conference on Cyberworlds (CW)*. IEEE. 2015, pp. 229–236.

[499] Hui Zhang. "Head-mounted display-based intuitive virtual reality training system for the mining industry". In: *International Journal of Mining Science and Technology* 27.4 (2017), pp. 717–722.

[500] Zhenqiu Zhang et al. "Head pose estimation in seminar room using multi view face detectors". In: *International Evaluation Workshop on Classification of Events, Activities and Relationships*. Springer. 2006, pp. 299–304.

[501] Liang Zhao, Gopal Pingali, and Ingrid Carlbom. "Real-time head orientation estimation using neural networks". In: *Proceedings. International Conference on Image Processing*. Vol. 1. IEEE. 2002, pp. I–I.

[502] Zheng Zhao, Yuchuan Wang, and Shengbo Fu. "Head movement recognition based on Lucas-Kanade algorithm". In: *2012 International Conference on Computer Science and Service System*. IEEE. 2012, pp. 2303–2306.

[503] Xianjun Sam Zheng et al. "Eye-wearable technology for machine maintenance: Effects of display position and hands-free operation". In: *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. 2015, pp. 2125–2134.

[504] Xiao Wei Zhong, Pierre Boulanger, and Nicolas D Georganas. "Collaborative augmented reality: A prototype for industrial training". In: *21th Biennial Symposium on Communication, Canada*. 2002, pp. 387–391.

[505] Tianyu Zhou, Qi Zhu, and Jing Du. "Intuitive robot teleoperation for civil engineering operations with virtual reality and deep learning scene reconstruction". In: *Advanced Engineering Informatics* 46 (2020), p. 101170.

[506] Zoran Zivkovic et al. "Improved adaptive Gaussian mixture model for background subtraction." In: *ICPR (2)*. Citeseer. 2004, pp. 28–31.

[507] Stefanie Zollmann et al. "Augmented reality for construction site monitoring and documentation". In: *Proceedings of the IEEE* 102.2 (Jan. 2014), pp. 137–154. DOI: 10.1109/JPROC.2013.2294314.

This Ph.D. thesis has been typeset by means of the TEX-system facilities. The typesetting engine was pdfLATEX. The document class was `toptesi`, by Claudio Beccari, with option `tipotesi=scudo`. This class is available in every up-to-date and complete TEX-system installation.