Doctoral Dissertation
Doctoral Program in Computer Engineering (29^{th}cycle)

Visualization and Human-Machine Interaction

By

Federico Manuri
******

Supervisor(s):
Prof. Andrea Sanna

Doctoral Examination Committee:
Prof. Fabrizio Lombardi , Referee, Northeastern University
Prof. Carlo Alberto Boano, Referee, Graz University of Technology
Prof. Giancarlo Iannizzoto, Università of Messina
Prof. Christian Micheloni, Università di Udine
Prof. Paolo Montuschi, Politecnico di Torino

Politecnico di Torino
2017

Declaration

I hereby declare that, the contents and organization 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.

Federico Manuri
2017

* This dissertation is presented in partial fulfillment of the requirements for **Ph.D. degree** in the Graduate School of Politecnico di Torino (ScuDo).

I would like to dedicate this thesis to my loving parents

Acknowledgements

I would like to thank my thesis advisor prof. Andrea Sanna of the Department of Control and Computer Engineering at Politecnico di Torino. The door to Prof. Sanna's office was always open whenever I ran into a trouble spot or had a question about my research or writing. He consistently allowed this thesis to be my own work, but steered me in the right direction whenever he thought I needed it.

I would also like to thank all the people that worked with me during these three years: dr. Andrea Stifanelli, prof. Fabrizio Lamberti, assistant prof. Gianluca Paravati, and my fellow PhD student Giovanni Piumatti. Without their passionate participation and input, I would not be able to complete my research work.

Finally, I must express my very profound gratitude to my parents and to my girlfriend for providing me with unfailing support and continuous encouragement throughout my years of PhD and through the process of writing this thesis. This accomplishment would not have been possible without them. Thank you.

Federico Manuri

Abstract

The digital age offers a lot of challenges in the field of visualization. Visual imagery has been effectively used to communicate messages through the ages, to express both abstract and concrete ideas. Today, visualization has ever-expanding applications in science, engineering, education, medicine, entertainment and many other areas.

Different areas of research contribute to the innovation in the field of interactive visualization, such as data science, visual technology, Internet of things and many more (Figure 1). Among them, two areas of renowned importance are Augmented Reality and Visual Analytics.

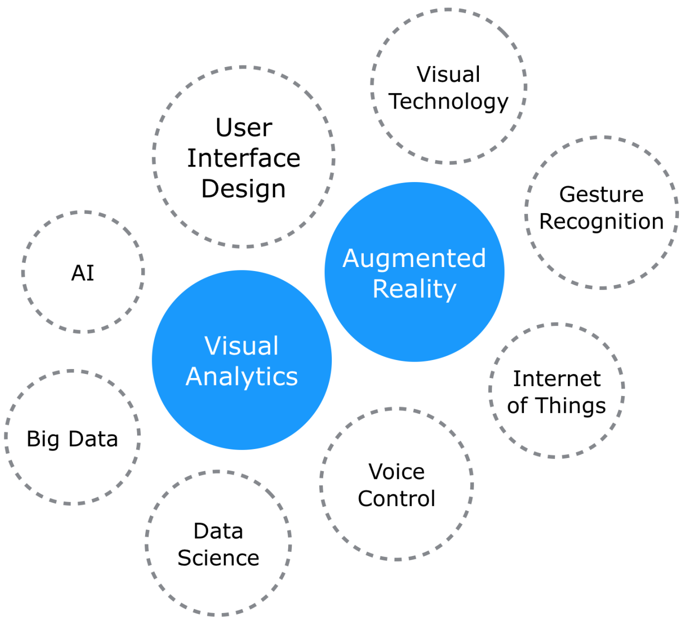

Fig. 1 Most important areas of research in the field of interactive visualization, as depicted by the Visual Analytics Lab of the Massachusetts Institute of Technology [1].

This thesis presents my research in the fields of visualization and human-machine interaction. The purpose of the proposed work is to investigate existing solutions in the area of Augmented Reality (AR) for maintenance. A smaller section of this thesis presents a minor research project on an equally important theme, Visual Analytics. Overall, the main goal is to identify the most important existing problems and then design and develop innovative solutions to address them.

The maintenance application domain has been chosen since it is historically one of the first fields of application for Augmented Reality and it offers all the most common and important challenges that AR can arise, as described in chapter 2. Since one of the main problem in AR application deployment is reconfigurability of the application, a framework has been designed and developed that allows the user to create, deploy and update in real-time AR applications. Furthermore, the research focused on the problems related to hand-free interaction, thus investigating the area of speech-recognition interfaces and designing innovative solutions to address the problems of intuitiveness and robustness of the interface.

On the other hand, the area of Visual Analytics has been investigated: among the different areas of research, multidimensional data visualization, similarly to AR, poses specific problems related to the interaction between the user and the machine. An analysis of the existing solutions has been carried out in order to identify their limitations and to point out possible improvements. Since this analysis delineates the scatterplot as a renowned visualization tool worthy of further research, different techniques for adapting its usage to multidimensional data are analyzed. A multidimensional scatterplot has been designed and developed in order to perform a comparison with another multidimensional visualization tool, the ScatterDice.

The first chapters of my thesis describe my investigations in the area of Augmented Reality for maintenance. Chapter 1 provides definitions for the most important terms and an introduction to AR. The second chapter focuses on maintenance, depicting the motivations that led to choose this application domain. Moreover, the analysis concerning open problems and related works is described along with the methodology adopted to design and develop the proposed solutions. The third chapter illustrates how the adopted methodology has been applied in order to assess the problems described in the previous one. Chapter 4 describes the methodology adopted to carry out the tests and outlines the experimental results, whereas the fifth chapter illustrates the conclusions and points out possible future developments.

Chapter 6 describes the analysis and research work performed in the field of Visual Analytics, more specifically on multidimensional data visualizations.

Overall, this thesis illustrates how the proposed solutions address common problems of visualization and human-machine interaction, such as interface design, robustness of the interface and acceptance of new technology, whereas other problems are related to the specific research domain, such as pose tracking and reconfigurability of the procedure for the AR domain.

Contents

List of Figures

List of Tables

Glossary

Acronyms / Abbreviations

$2D$ Bidimensional

$3D$ Three-dimensional

AMF Augmented Maintenance Framework

AR Augmented Reality

$AVOS$ Automatic Vocal Scanning

$BVOS$ Bidirectional Vocal Scanning

CAD Computer-Aided Drawing

$CAVE$ Cave Automatic Virtual Environment

CSS Cascading Style Sheet

FN False Negative

FP False Positive

FPS Frames Per Second

$FTUE$ First Time User Experience

GPS Global Positioning System

GUI Graphic User Interface

HAR Handheld Augmented Reality

HCI Human-Computer Interaction

HMD Head Mounted Device

HMPD Head-Mounted Projective Display

HTML HyperText Markup Language

HUD Head-Up Display

ICT Information and Communication Technology

IETM Interactive Electronic Technical Manuals

IIF Intuitive Interface Framework

IP Internet Protocol

IVOS Inverse Vocal Scanning

KB KiloByte

LCD Liquid-Crystal Display

MAR Mobile Augmented Reality

MSCW Must Should Could Would

MW Multi-word

POI Point Of Interest

QRCode Quick Response Code

RFID Radio-Frequency IDentification

RGB Red Green Blue

RSD Retinal Scanning Displays

SC Semantic Cause

SVG Scalable Vector Graphics

TC Technological Cause

TCP Transport Control Protocol

TN True Negative

TP True Positive

UI User Interface

UI User Interface

VPN Virtual Private Network

VR Virtual Reality

VRD Virtual Retina Display

$VRML$ Virtual Reality Modeling Language

WCS World Coordinate System

XML Xperimental Markup Language

Chapter 1

Introduction

Part of the work described in this chapter has also been previously published in [2], [3] and [4]

Nowadays, the usage of computer systems is widespread to many kinds of job, hobbies and other common activities. Two important factors that affect the usage of computers are the interaction with the machine and the comprehension of the information provided by such usage. Both of them have been quite affected in the last years by the innovations in the field of Augmented Reality (AR). The first part of this research focuses on investigating human-machine interaction through AR solutions, with regard to both the interface and the information visualized.

This chapter is organized as follow: section 1.1 provides some definitions of the most important terms. Section 1.2 proposes a brief history of Augmented Reality. The architecture of a generic Augmented Reality framework is described in section 1.3 whereas AR technologies are presented in section 1.4. Section 1.5 depicts the multitude of application domains in which Augmented Reality is effectively adopted.

1.1 Definitions

1.1.1 Visualization

The term visualization (or visualisation) groups all the techniques adopted to create images, animations or diagrams with the purpose of communicating a message. The usage of visual imagery to communicate messages has been effectively used through the ages to express both abstract and concrete ideas. Nowadays, computer graphics is largely used to create visualizations, but, in the beginning, the lack of graphic complexity limited its usefulness.

Starting from 1987, with the publication of 'Visualization in Scientific Computing', many conferences and workshops have been held on the topic by both the IEEE Computer Society [5] and ACM [6]. Moreover, many of the principles related to visualization have been thoroughly analyzed and described by Edward Tufte through three books published at the end of the last century: 'The Visual Display of Quantitative Information' (1983), 'Envisioning Information' (1990) and 'Visual Explanations: Images and Quantities, Evidence and Narrative' (1997).

1.1.2 Human-Machine Interaction

The term human-machine interaction defines a branch of computer science that researches both physical and visual interaction interfaces between people and machines. Researchers in this field investigate novel ways that improve and simplify the human interaction with machines, taking into account many fields of research related to communication, such as behavioral science, interface design and several other fields of study. A popular branch of Human-Machine Interaction is Human-Computer Interaction or HCI, which focuses on the interaction with computer systems.

The interaction paradigm relays on two combined components: the physical interface used by the user and the interface (usually graphical) used by the computer to provide the feedback. The first one involves all kind of interaction devices such as the keyboard, the mouse, game-pads, touch screens, microphones and many other innovative solutions. The second one defines how the machine provides a feedback to the user: usually it involves the use of a display, in which case the term graphical user interface is used. In other cases, if a visualization device is not available, an audio interface could be used such as the ones adopted by call centers or the audio guides available in most museums. More unusual solutions may involve other senses, such as interfaces with a haptic feedback.

Even if the first known use was in 1975 [7], Human-Computer Interaction became popular in 1983 thanks to a book published with the title of 'The Psychology of Human-Computer Interaction' [8]. This book delineates the main feature that denotes the action of using a computer as an interaction: compared to other tools which offer only limited uses, a computer has many uses and this takes place as an open-ended dialog, since the computer responds to the user requests, and vice versa.

1.1.3 Augmented Reality

The term Augmented Reality refers to a set of devices and technologies that grant the user the ability to see both the real world and the virtual space at the same time, thus enhancing the user perception of reality. Whereas Virtual Reality (VR) has the goal of creating entire virtual worlds, in an AR system virtual object are displayed with the purpose of coexisting with the real ones. An augmented space ‘blends’ together real and virtual objects, positioning and aligning the virtual elements with the purpose of making them realistic: from the user point of view, the virtual elements should seem as part of the real world. Since artificial and physical objects are mixed together, the user can move in a hybrid space without constraints.

On the reality-virtuality continuum defined by Milgram and Kishino, AR is part of the general area of mixed reality [9]. Both virtual environments and augmented virtuality, in which real objects are added to virtual ones, replace the surrounding environment by a virtual one. In contrast, AR provides local virtuality.

Relation with Virtual Reality

Computer generated part of augmented environment makes AR similar to the concept of virtual reality, but with some important differences. In the essay ‘A Survey of Augmented Reality’ of 1997 Ronald T. Azuma asserts that:

“*Augmented Reality (AR) is a variation of Virtual Environments (VE), or Virtual Reality as it is more commonly called. VE technologies completely immerse a user inside a synthetic environment. While immersed, the user cannot see the real world around him. In contrast, AR allows the user to see the real world, with virtual objects superimposed upon or composited with the real world. Therefore, AR supplements reality, rather than completely replacing it. Ideally, it would appear to the user that the virtual and real objects coexisted in the same space, similar to the effects achieved in the film ‘Who Framed Roger Rabbit?’*” [10].

In essence, the two realities, the augmented and the virtual one, are so distinguished:

- in virtual reality, information that is added or subtracted is predominant, to the extent that persons find themselves immersed in a situation in which the natural perceptions of the five senses do not seem to be present anymore and are replaced with the artificial ones;
- in augmented reality, instead, the person has the sensation to live the common physical reality, even taking advantage of additional information or manipulations of that. Therefore, it does not tend to separate the two worlds, real and virtual, but to join and mix them together.

Moreover, virtual reality presents a number of issues related to health and safety: first of all, long-term effects of virtual reality on vision and neurological development are still unknown. Furthermore, users may become disoriented in a purely virtual environment and suffer balance issues; if the interaction is based upon movement and the user is not confined to a limited area, navigation may even becomes dangerous without external sensory information. Finally, there have been rising concerns that some users may experience virtual reality addiction [11].

The distinction between VR and AR is therefore forced: the mediated reality, in fact, can be considered ad a continuum, in which VR and AR take place adjacent and are not simply two opposite concepts [12].

Usage

The main purpose of an AR-based system is to allow users to 'experience' the real world enriched by a set of overlapping computer-generated contents and eventually other sensory inputs, such as audio augmentation through earplugs or speakers [13]. The augmentation generated by the computer is strictly related to the user context, both in terms of view and location: the virtual objects are aligned to the real ones meaningfully to provide the user a better understanding of the surroundings. As augmented reality is something strictly related to our experience of the real world, an AR system should collect a wide range of information from different sensors (e.g. camera, GPS, and so on). Another peculiarity of AR systems is to be real time, so the AR hardware device should provide enough computing power to interactively run an AR application.

When interacting with an AR application, the user can see the real world directly or can observe it through a camera. This mediated reality pervades our everyday life: study, training, work, relaxation are just some occasions in which the user can benefit of AR applications. Even if it is not easy to exactly define what augmented reality is and what is not, it is generally acknowledged to denote an application as an 'augmented reality application' when it displays a set of computer-generated information (usually identified as assets) overlaid to the real world. The category of augmented reality assets consist of text labels, audio tracks, 3D models, animations and videos. Even if AR may involve all the senses, the most used type of asset consist of computer-generated virtual objects that need to be presented to the user.

An indisputable advantage that can be attributed to AR is that the user of AR applications can keep contact with the real world. This is an advantage for two reasons: firstly, part of the space that the user sees already exist, so it is not necessary to represent it through a computer-generated model of it; secondly, the user physical point of view is preserved, avoiding the physical and mental annoyance that a detachment from the real world can produce. With his capability of bridging the gap between real and virtual worlds, Augmented Reality can be considered the best solution every time it is necessary to represent real and computer-generated elements within the same space.

1.2 A Brief History of Augmented Reality

Nowadays Augmented Reality (AR) is a renowned concept but even if its origins can be dated back to the sixties, it has been formalized only in 1994 by Milgram and Kishino [9] as the relationship between real space, virtual space and all the intermediate forms of mixed space. Sutherland was the first to describe Augmented Reality back in 1965, when he wrote in 'The Ultimate Display':

"There is no reason why the objects displayed by a computer have to follow the ordinary rules of physical reality with which we are familiar. The kinesthetic display might be used to simulate the motions of a negative mass. The user of one of today's visual displays can easily make solid objects transparent - he can 'see through matter'!" [14]

Only three years later, in 1968, Sutherland proposed the first acknowledged AR prototype, based on a Head Mounted Device (HMD) [15]. The prototype, designed with his students at Harvard University, consisted of an helmet equipped with a digital display for immersive reality. The user could wear the helmet to observe the room and see virtual information displayed over the physical world. Modern Head Mounted Devices are still based on that prototype, even if efficiently modified, and represent one of the most common and notorious device used to enjoy AR application. The original HMD had not so much in common with the definition of 'immersive reality' since the helmet was so heavy that could not be worn without the support of a mechanical arm that held the device on top of the user. Since the arm was attached to the ceiling of the laboratory, the daunting sight of the machinery lead to a memorable nickname: 'The Sword of Damocles'.

After Sutherland, more than twenty years passed before the term Augmented Reality was officially coined. Tom Caudell and David Mizell, two scientists employed at Boeing Corporation, developed an experimental AR system with the purpose of simplifying the manufacturing process of the air company [16]. The AR was adopted to enhance the assembly process, showing the workers where and how to lay wiring harnesses. In the same year, Steven Feiner, Blair MacIntyre and Dorée Seligmann developed another augmented reality system, the KARMA (Knowledge-based Augmented Reality for Maintenance Assistance) with the purpose of assisting maintainers involved in maintenance tasks [17]. The system displayed wireframe schematics on top of the components that need to be repaired, together with the instructions needed to perform the repair procedure. Even if these solutions were distant from true mobile solutions, soon enough the technological improvements in computing and tracking devices allowed graphical overlay on mobile devices.

By the end of the century, AR became a renowned field of research, with numerous conferences on the topic such as the International Symposium on Mixed Reality and the International Symposium on Augmented Reality. In 1999, Feiner proposed the first prototype of a mobile AR system, the MARS (Mobile Augmented Reality System): the purpose of the system was to provide tourist information and representations of ancient buildings, displayed on a 'portable' head-mounted device through AR [18]. In the same year, the ARToolKit was publicly disclosed at SIGGRAPH, the first time a working AR system had been seen outside of research labs in the US, heralding the start of the AR industry, released as open source software after only two years. Different research groups were founded in those years such as the ARVIKA consortium in Germany [19] and the Mixed Reality Systems Laboratory (MRLab) in Nottingham [20].

In 2001 a survey by Azuma et al. [21] identified the three properties that characterize an AR-based system: combination in a real environment of both real and virtual objects, alignment of the virtual objects with respect to the real ones and interactive execution of the system. In the same year, the first International Symposium on Mixed and Augmented Reality (ISMAR) was organized, which would become year after year the major symposium for industry and research focused on AR, with the purpose of proposing an international stage for exchanging information, problems and solutions on the matter [22]. This increasing interest in AR technologies and its widespread in the following years could be explained through the Technology Acceptance Model (TAM) proposed by Davis on 1989 [23] and its evolutions, that clearly points out how the perceived usefulness and ease of use are at the foundations for the acceptance and adoption of any new technology. The turning point for the spread and acceptance of AR systems has been represented by the technological improvements in the field of mobile devices.

Until some years ago, the lack of cost-affordable devices was the main barrier to a wider adoption of AR applications. Nowadays, the widespread adoption of mobile devices has removed this limitation, as smartphones and tablets feature all the sensors and processing units needed to develop and deploy AR applications. Moreover, the technology innovations that affect mobile devices can produce new challenging products, commonly referred to as wearables, and industries are moving forward with new categories of AR devices, such as the Google Glass project [24], Microsoft Hololens [25] and contact lenses from Innovega [26]. The global market for augmented reality is growing fast and the pervasive adoption of AR technologies implies an undeniable impact on the society.

It has been a long journey from that first prototype in 1968 and now AR is widespread in everyday life. AR applications are commonly used for many purposes such as working, studying, playing and in many other situations. At the same time, complex technology that were designed for very peculiar scopes, such as industry or military, are now available for commercial applications. AR is well consolidated and a huge number of applications for smartphones, tablets and other innovative user interfaces are available for a wide range of applications.

1.3 AR Architecture

AR systems are characterized by at least three blocks: a tracking system, an asset/scene generator and a combiner.

1.3.1 Tracking System

Modern AR systems usually rely on optical trackers: a camera is used to frame the real environment and then the video stream is analyzed by the tracking module to compute the position and orientation of the head with the respect to the framed objects. Position and orientation are essential to show the virtual objects to the user correctly aligned with the real ones.

When the tracking system needs to rely on an absolute reference system, it is necessary to provide a reference frame to the tracking system. This is the case when the tracking is object dependent and unrelated from an absolute value associated to the real world, such as GPS coordinates. If a reference frame is needed, two are the available approaches: marker-based and marker-less.

Marker-based solutions require the camera to frame an explicit reference to define a World Coordinate System (WCS), which means that a reference object has to be intentionally positioned in the environment. Commonly, QR codes or AR tags are used to this purpose. Since the size of the marker is known, the tracker can compute the exact and absolute position of the camera with respect to the WCS.

On the other hand, marker-less approaches make use of environment features as reference frame, such as a building or another object already present in the environment. Recent solutions offer the chance to initialize the coordinate system from an image of the reference object or even from a 3D model. This alternative allows computing relative positions but the system should already know the real size of the reference object to correctly compute the position of the user. Innovative solutions also involve the usage of accelerometers to provide a more precise alignment if the three axis information is originally available for the reference image.

If it is not possible to feed the tracking system through optical devices, for example if the AR glasses do not include a camera or the computational power is not enough to process the video from the camera in real time, other tracking solutions can be used: inertial, mechanical and magnetic. This kind of solutions are not very much widespread, but more information is available in 'Motion tracking requirements and technologies' [27].

1.3.2 Scene Generator

When the position of the user is defined, the tracking module sends this information to the scene generator block. Depending on the head's location and orientation, the scene generator computes the assets visible for that specific frame and accurately aligns them; obviously this is not the case for audio messages or for graphical information that are not related to a specific position in the real world, but that will appear as an overlaid graphical interface, such as a menu or a text message.

1.3.3 Combiner

The combiner is the last block and its purpose is to overlap the assets to the user view; the combiner acts in different ways according to the used AR paradigm.

See-through Devices

The first paradigm is based on see-through devices. A see-through device is identified by a screen that is partially transparent, thus allowing the user to see the physical environment with his/her own eyes (Figure 1.1 left). When see-through devices are used, assets and real objects are mixed together by an optical combiner. The assets generated by the scene generator are overlapped by an optical effect, projecting them on the see-through screen. This solution allows the user to directly observe the world around him/her, but it requires special purpose devices such as AR glasses. This kind of glasses can be both monocular or binocular, in which case the user will be able to correctly visualize also 3D assets. The assets can be projected over the lenses or displayed on semitransparent mini-monitors placed between eyes and lenses. Usually AR glasses include a RGB camera that directly feeds the tracking system.

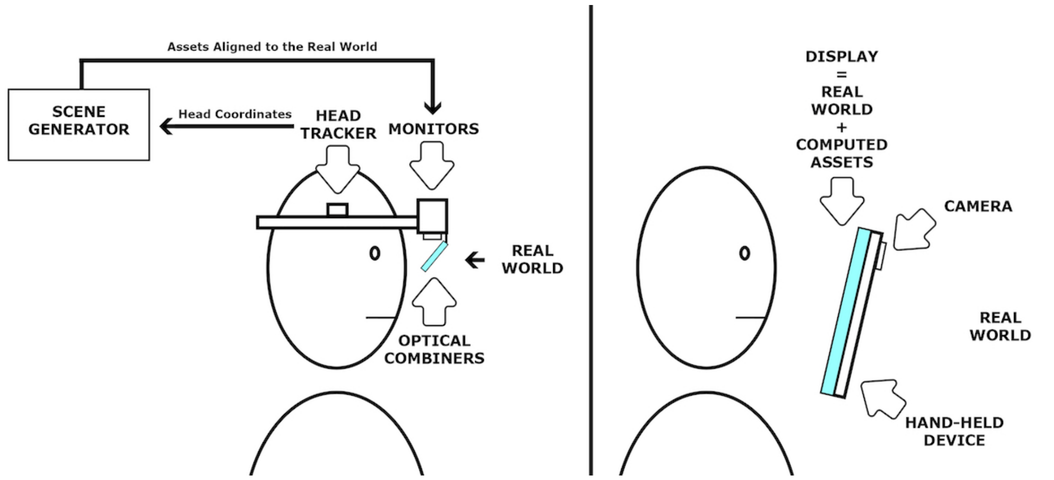

Fig. 1.1 The see-through paradigm (left): the user is able to see the real world with his/her own eyes and assets are overlapped by an optical combiner. The hand-held paradigm (right): the user perceives the real world through the video streaming coming from the camera and assets are overlapped by a video combiner.

Hand-held Devices

The second paradigm is based on hand-held (mobile) devices (see Figure 1.1 right). Mobile devices (e.g. tablets and smartphones) encompass all the hardware necessary to implement an AR system: the camera to gather a video of the real world, a display to show the augmented environment and the computational power to compute the camera position-orientation (by optical tracking), generate assets and combine virtual objects with the video. Mobile AR (MAR) has become very popular as cellular phones and personal digital assistants have been replaced by smart devices able to run computational intensive apps. With respect to the see-through paradigm, the MAR approach presents users a mediated reality where a direct perception of the surrounding world is not available. Assets and video of the virtual world are mixed together by a video combiner module.

Mobile Devices

The third paradigm, called monitor-based AR, is conceptually comparable with the previous one, but camera, computational power and display (a monitor) are not encompassed in a single device. This paradigm is used when either large displays are needed or the camera must be independent (e.g. in augmented endoscopy).

1.3.4 Limitations

This high level description of an AR system clearly outlines possible challenges AR applications have to tackle. First of all, AR systems have to deal with performance, mainly concerning the real time computation of position and orientation of the camera. Then, a second issue concerns the precision of asset alignment: several applications can require a sub-millimeter accuracy (e.g. medical AR applications). The third issue is related to the user interface: users have to be able to interact with augmented contents, but the traditional keyboard-mouse paradigm is not generally available. The last two challenges concern mobility and visualization; nowadays mobile devices seem to fully satisfy the best part of application fields. On the other hand, see-through visualization devices are not still fully compliant to contrast, resolution, brightness and field of view requirements.

1.4 AR Technologies

1.4.1 Interfaces

Among human sensory inputs, sight, hearing and touch are currently the senses that AR systems normally apply. This section mainly focuses on visual displays, however aural (sound) displays and haptic (touch) displays are mentioned briefly below.

Sound Displays

In the augmented reality field, aural display applications are mainly limited to headphones and loudspeakers. True 3D aural display is still found in experimental stages of immersive simulations of virtual environments. ‘Haptic audio’, in fact, refers to sound that is felt rather than heard [28] and is already applied in consumer devices to increase the sense of realism and impact, but also to enhance user interfaces of mobile phones [29]. Recent developments in this area are presented in activities such as the international workshop on Haptic Audio Visual Environments and the international workshop on Haptic and Audio Interaction Design.

Haptic Displays

Haptics is like teleoperation, but the remote slave system is purely computational. Haptic devices are in effect robots with a single task: to interact with humans [30]. The haptic sense is differed into the kinesthetic sense (force that moves) and the tactile sense (touch). Force feedback devices like joysticks and steering wheels are well known among gamers, whereas a popular haptic device in medical operation and other areas is the Phantom: it optionally provides interaction through a pinch or scissors extension. Tactile feedback devices can also communicate parameters such as roughness, rigidity, and temperature: data gloves use various technologies to sense and are very reliable, flexible and widely used in VR for gesture recognition. However, in AR they are suitable only for brief, casual use, as they hinder the use of hands in real world activities and are somewhat awkward in a more general application.

Visual Displays

There are essentially three ways to visually present augmented reality. Closest to virtual reality is video see-through, where the virtual environment is replaced by video of reality and the AR is overlaid upon the digitized frames. Another way that includes Sutherland's approach is optical see-through and leaves the real-world perception unaltered but displays only the AR overlay by means of transparent mirrors and lenses. The third approach is to project AR information onto real objects themselves resulting in projective displays.

1.4.2 Visual Displays Technologies

Video See-through

In spite of being the cheapest and easiest to implement, this display technique offers various advantages. Since reality is digitized, it is easier to mediate or remove objects from it. This includes removing or replacing fiducial markers or placeholders with virtual objects. Also, brightness and contrast of virtual objects are matched easily with the real environment: evaluating the light conditions of a static outdoor scene is of importance when the computer-generated content has to blend smoothly with the environment [31]. In addition, the digitized images allow tracking of head movement for better registration and make possible to match perception delays of the real and virtual world. On the other hand, disadvantages of video see-through include a low resolution of reality, a limited field-of-view (although this can easily be increased), and user disorientation for parallax effect due to the camera's positioning at a distance from the viewer's true eye location, causing significant adjustment effort for the viewer [32]. This problem was solved at the MR Lab by aligning the video capture [33]. A final drawback is the focus distance of this technique that is fixed in most display types, providing poor eye accommodation. Some head-mounted setups can however move the display (or a lens in front of it) to cover a range of 25 meters to infinity within 3 seconds [34]. Like the parallax problem, binocular displays (where both eyes see the same image) cause significantly more discomfort than monocular or binocular displays, both in eyestrain and fatigue [35].

Optical See-through

These displays not only leave the real-world resolution intact, they also have the advantage of being cheaper, safer, and parallax-free (no eye-offset due to camera positioning) than video see-through solutions. Optical techniques are safer because users can still see when power fails, making this an ideal technique for military and medical purposes. However, other input devices such as cameras are required for interaction and registration. Also, combining the virtual objects holographically through transparent mirrors and lenses creates disadvantages as it reduces brightness and contrast of both the images and the real-world perception, making this technique less suited for outdoor use. The field-of-view is limited for this technique and may cause clipping of virtual images at the edges of the mirrors or lenses. Finally, occlusion (or mediation) of real objects is difficult because their light is always combined with the virtual image. Kiyokawa solved this problem for head-worn displays by adding an opaque overlay using an LCD panel with pixels that opacify areas to be occluded [36]. Virtual retinal displays or retinal scanning displays (RSDs) solve the problems of low brightness and low field-of-view in optical see-through displays. A low-power laser draws a virtual image directly onto the retina which yields high brightness and a wide field-of-view. RSD quality is not limited by the size of pixels, but only by diffraction and aberrations in the light source, making very high resolutions possible as well. Together with their low power consumption, these displays are well-suited for extended outdoor use. Still under development at Washington University and funded by the U.S. military, current RSDs are mostly monochrome (red only) and monocular (single-eye) displays. Schowengerdt already developed a full-color, binocular version with dynamic refocus to accommodate the eyes that is promised to be low-cost and light-weight [37].

Projective

These displays have the advantage that they do not require special eye-wear thus accommodating user's eyes during focusing, and they can cover large surfaces for a wide field-of-view. Projection surfaces may range from flat, plain colored walls to complex scale models [38]. However, as with optical see-through displays, other input devices are required for interaction. Also, projectors need to be calibrated each time the environment or the distance to the projection surface changes (crucial in mobile setups) but, fortunately, calibration may be automated using cameras, for example a multi-walled automatic virtual environment (CAVE) with irregular surfaces [39]. Furthermore, this type of display is limited to indoor use only due to low brightness and contrast of the projected images. Occlusion or mediation of objects is also quite poor, but for head-worn projectors this may be improved by covering surfaces with retro-reflective material. Objects and instruments covered in this material will reflect the projection directly towards the light source that is close to the viewer's eyes, thus not interfering with the projection.

1.4.3 Visual Displays Configurations

AR displays may be also classified into three categories based on their position between the viewer and the real environment: head-attached, hand-held, and spatial.

Head-attached

Visual displays attached to the head include the video/optical see-through head-mounted display (HMD), virtual retinal display (VRD), and head-mounted projective display (HMPD). A current drawback of head-worn displays is the connection to computers, thus even adopting laptops the mobility would be restricted due to limited battery life [40]. Battery life may be extended by moving computation to distant locations (clouds) and provide wireless connections using standards such as Wi-Fi or Bluetooth.

Hand-held

This category includes hand-held video/optical see-through displays as well as hand-held projectors. Although this category of displays is bulkier than head-worn displays, it is currently the best work-around to introduce AR to a mass market due to low production costs and ease of use. For instance, hand-held video see-through AR acting as magnifying glasses may be based on existing consumer products like mobile phones with navigation information [41].

Spatial

The last category of displays is placed statically within the environment and includes screen-based video see-through displays, spatial optical see-through displays, and projective displays. These techniques lend themselves well for large presentations and exhibitions with limited interaction. Early ways of creating AR are based on conventional screens (computer or television) that show a camera feed with an AR overlay: this technique is now being applied in the world of sports television where environments such as swimming pools and race tracks are well defined and easy to augment. Head-up displays (HUDs) in military cockpits are a form of spatial optical see-through and are becoming a standard extension for production cars to project navigational directions in the windshield [42].

1.5 AR Applications Domains

At the beginning, AR was adopted in six distinct areas: manufacturing and repair, medicine, robot path planning, annotation and visualization, entertainment and military applications. Later on, AR applications reached other important domains such as education, cultural heritage, architecture, tourism, advertising and many others. This analysis well depicts how AR is pervasive in people's everyday life: everybody may have to cope with AR applications when working, playing, traveling, studying and in many other activities.

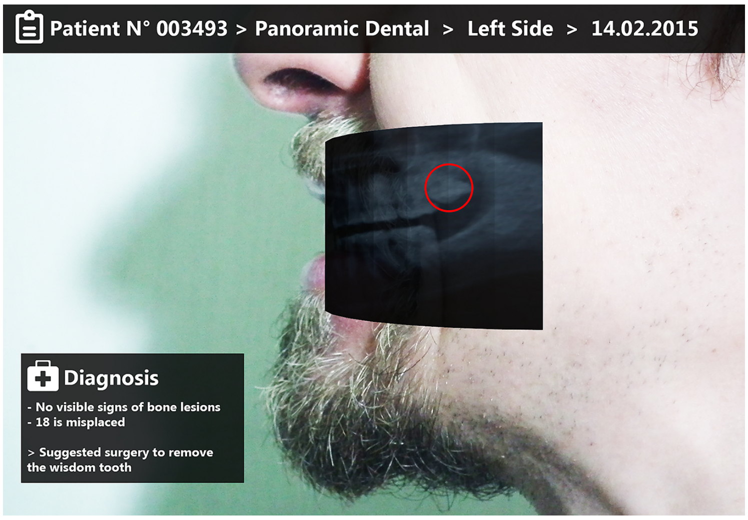

Fig. 1.2 Medical application for dental surgery. The application shows the medical record for a patient, overlapping the radiography image to the face; points of interest are highlighted as well as doctor's annotations.

1.5.1 Medicine

As their natural attitude is to bridge the gap between real and virtual, AR technologies were immediately identified as a valuable tool to bring patients and their medical data into the same space. The potential of AR in medical applications was foreseen by Steinhaus (an Austrian mathematician) in 1938 [43]: Steinhaus suggested a method to display a bullet inside a body by a very cumbersome overlay process. On the other hand, the first real application of AR in medicine can be dated back to 1986, when a system to integrate data from computer tomography into an operating microscope was proposed. Readers interested in examining in depth applications of AR in medicine can refer to [44].

Recent advances in medical imaging have provided scientists and physicians with a huge amount of data that could be profitably used to support several activities. Anatomical and functional data can be a support in surgery as well as in diagnostic of preoperative and intraoperative data or in training tasks. The use of AR in surgery is strictly related to the display technology the surgeon opts for: computer-generated assets can be directly overlapped onto the operating microscope or can be displayed over a monitor (augmented endoscopes can be considered as a particular case of monitor-based AR). Figure 1.2 shows an example of medical application: the patient's medical record is shown and the radiography image is overlapped to the face with precise positioning. The projection of assets directly on the patient is apart from a particular visualization technology, but it involves very complex setups, which need accurate calibrations.

Much more than other application fields, AR in medicine has to overcome three main issues: tracking precision, misperception and interaction with synthetic data. The precision required for several surgical operations is of the sub-millimeter order, therefore assets must be overlaid very accurately. On the other hand, medical AR generally involves very limited and controlled working indoor volumes and current tracking systems are able to provide the required precision under these conditions. The misperception is basically related to a wrong perception of depth (although assets are correctly aligned, the user perceives them in a wrong position), but this issue can be mitigated by using stereoscopic visualization devices. The interaction issue is more generally related to user interface design problems; for instance, a surgeon cannot interact with assets by using touch, therefore natural and multimodal user interfaces have to be implemented. Multimodal interfaces allow the user to choose among different input modalities: gesture/pose recognition and speech recognition can be two alternatives to naturally interact with computer-generated contents. Unfortunately, these alternative input modes can introduce robustness issues, which have to be taken into account when safety-critical systems such as medical ones are designed.

1.5.2 Assembly, Maintenance and Repair

Technicians involved in complex maintenance and repair tasks often need to refer to instruction manuals to correctly complete assigned procedures. This might entail a high cognitive load deriving from a continuous switch of the attention between the device under maintenance and the manual. In other words, mistakes are more likely and repair times (and hence the costs) can grow up. A first attempt to mitigate these issues was the introduction of Interactive Electronic Technical Manuals (IETMs) able to replace paper instructions. On the other hand, also IETMs cannot be completely part of the interaction between technician and equipment to be maintained. AR can efficiently tackle this issue and manufacturing-repair has been immediately identified as one of the most promising application field of AR [10].

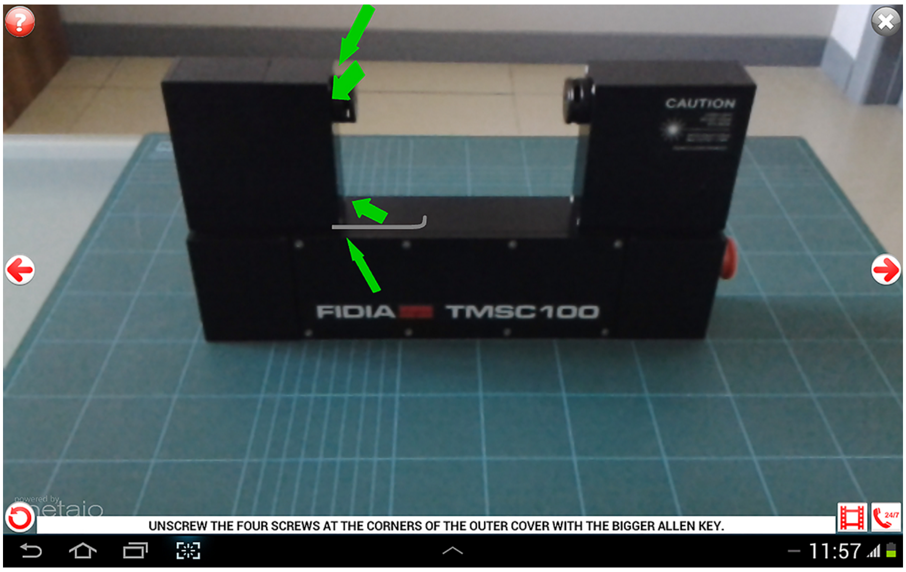

Fig. 1.3 AR application for maintenance procedures. The application supports the technician through the repair task of an industrial device; textual and 3D assets are shown.

Assets are overlaid and correctly aligned with respect to the device to be maintained and can be conveyed to technicians while they are performing the procedure. Moreover, AR applications for maintenance and repair are often completed by telepresence systems; in this way, a remote expert can interactively support maintainers when AR aids are not sufficient. Figure 1.3 provides a technician's view of an AR maintenance application with the head-mounted paradigm: audio, 3D model animated and textual assets describe the instructions the technician should follow to perform the maintenance procedure.

Benefits of the AR support in maintenance, repair and assembly tasks have been deeply analyzed in [45]. First attempts of supporting technicians by AR tools can be dated back to 1990s and they were all based on special purpose hardware (e.g., HMDs). On the other hand, recent advances of mobile technologies have opened new challenging and intriguing scenarios for this type of applications. Tablets and smart-phones allow users to perform some tasks such as routine maintenance of a car, furniture assembly and installation of electrical appliances in an easier way. If we consider the huge amount of money related to these activities, it is immediately clear the tremendous impact the AR might have on our everyday life. At the present time, the spread of AR in maintenance and repair tasks is very limited; this is mainly due to the time needed to create, change and improve the procedures.

1.5.3 Entertainment and Sport

The entertainment industry is one of the most important drivers of ICT technology progress and a lot of improvements in augmented reality can be related to it. Video game players have a desire: they want to be part of the game. If the player is more involved in the game, then the game experience will be enhanced. This idea has inspired the design of all modern game consoles; in other words, players do not control characters but they are the characters. AR aims to bridge the gap between real and virtual, therefore it is the best tool to provide users a new game experience. The real world can become the set for a game (see for instance ARQuake [46]) and the player can experience a completely new and exciting game modality; moreover, beyond a more natural and intuitive interaction, AR-based games can be played, in general, everywhere, without any space limitation. Another important benefit concerns the development step: all game scenarios have to be completely modeled and rendered in a traditional game; on the other hand, AR games use the real world as a scenario and only virtual characters and assets have to be created. Figure 1.4 shows an example of this type of augmented game: a number of enemies is placed in the scenario and move towards the player, who has to shoot them down to gain points and to move to avoid the monsters and continue the game.

Modern game consoles also implement AR by using different types of cameras to augment computer graphics onto live footage. Also the market of stickers can take advantage from AR: when a sticker is framed by a personal device running a specific AR application, the user can play on the device multimedia contents about the star (e.g. a football player) or the personality depicted on the sticker. AR is often used for augmenting live broadcast of sport events [47]. For instance, computer-generated aids are added to the raw video images in order to show the off-side line in a football match or the trajectory of the puck in hockey games. Maybe, the most famous system for the augmentation of sport events is Hawk-Eye; Hawk-Eye is a system used in tennis matches, which provides the tracking and the visualization of balls trajectories, thus enabling players to challenge the Referees' decisions. In the augmentation of many sport events, assets have to be overlapped to raw images in real time; this means that the AR system has to be able to identify and track a given object in the scene in a very performing way. AR plays also a very important role when advertisements and logos have to be inserted in live video broadcasting: messages

Fig. 1.4 Entertainment application. The game requires the player to move in the real space in order to aim and shoot at the enemies; alien enemies are placed in the real environment.

as well as 3D objects/animations can be aligned to real objects framed by the set of different cameras used to film the event.

1.5.4 Collaborative Visualization Spaces

Data visualization is a very broad discipline encompassing several different fields; for instance, information visualization aims to find new paradigms to efficiently display huge amount of data (e.g. the network traffic over the Internet), whereas scientific visualization aims to present users phenomena that are very hard (or impossible) to perceive (e.g. air flows around to a plane wing). The intrinsic nature of AR provides visualization a worthwhile tool to display virtual objects within a physical space; moreover, AR is particularly suitable for collaborative visualization. Collaborating users communicate by using speech, gaze, gesture and other nonverbal cues. These 'communication channels' are often inhibited or limited when traditional collaborative work tools are adopted: tele-presence and screen-based collaboration often create barriers between physical and virtual space. On the other hand, virtual collaborative spaces migrate both objects and

Fig. 1.5 Collaborative application. The application allows users to visualize a virtual collaborative space and to add notes in order to share information with other users.

users (represented by avatars) in a cyberspace, thus altering usual communication channels. Face-to-face AR collaborative applications are able to avoid this problem, thus allowing users to communicate with each other by usual verbal and nonverbal cues within a physical space [48]. Moreover, AR applications can provide the users with custom visualizations, thus enabling custom views of the dataset. This is basically obtained by organizing the data to be displayed on layers: each layer should contain computer-generated objects, which share one or more attributes (e.g. they share the same material or they are sub-parts of the same object) and each layer can be activated/deactivated. A key role in collaborative work is often played by annotation: users have to be able to add their own textual comments to parts of the augmented space as well as they have to be able to share these annotations with other collaborating users. Figure 1.5 shows an example of collaborative application, based on a 3D model of a city's district: the users can share notes attaching them to the different parts of the 3D model (e.g. buildings, streets, squares).

AR applications for collaborative visualization often provide users tangible interfaces. A tangible interface links a real object to a computer- generated one; for instance, a building could be related to a physical object such as a small box: transformations applied to the box (e.g. translations and rotations) will be propagated to the associated virtual object. Of course, the tracking system of the AR application has to be able to track both users' heads and artifacts selected as alter egos of virtual objects; at this purpose, marker-based approaches are mainly used. Features of an AR collaborative visualization application are: virtuality (computer-generated objects can be displayed and examined), augmentation (virtual objects can be displayed in a physical space and real objects can be annotated), cooperation (several users can cooperate in a natural way), independence (each user can customize the dataset to be displayed) and individuality (each user can customize the graphic metaphors used to represent data). The first example of collaborative AR visualization is the Studierstube [49]. A main issue still concerns AR collaborative visualization tools: a better interaction form is obtained by using the see-through paradigm, but AR-glasses (e.g. binocular glasses and HMDs) often partially cover the user's eyes, thus inhibiting (or strongly limiting) gaze communication, which represents an important form of nonverbal cue. Monocular glasses can mitigate this issue, but they are not able to provide stereoscopic visualization of 3D objects.

1.5.5 Tourism

AR can play a key role in tourism mobilities [50] and a significant improvement has been obtained by new generation smartphones and tablets, which are often equipped with GPS sensors, and fast network connections. These devices are therefore able to support location-based AR services. A tourist experience can be enriched (and mediated) by adding multimedia and customized contents according to the tourist's needs [51]. Basically, three types of AR applications for tourism can be categorized. Augmented guides are the first type of AR application for tourism; an augmented guide searches, retrieves and visualizes information gathered from several Internet sources (e.g., touristic portals). Information are arranged in order to provide the users with all the support necessary to: organize travels, reserve tours, rent cars, and so on. The first example of an augmented guide is Tuscany+: the official augmented reality application of the Italian region Tuscany. Information gathered by the AR application can be also used to generate a sort of hybrid space: the physical space (e.g. a square of a city) is filled up with information coming from the cyber space (e.g. the Net). Figure 1.6 shows a Tourism application that provides the user with his current location and a set of nearby Points of Interest (POIs); an arrow near the POI tag indicates the direction to reach it; each POI is grouped by color and icon to simplify recognition. Several augmented guides allow the user to mark and share a

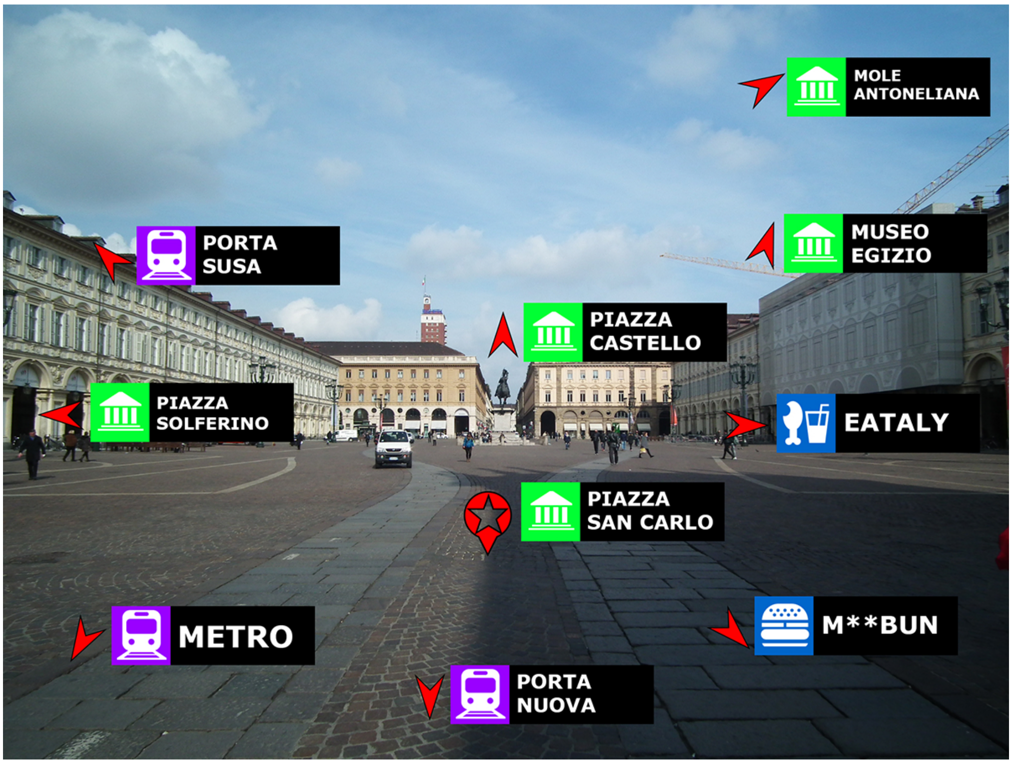

Fig. 1.6 Tourism application. The application shows the user's location and a selection of points of interest in the nearby area, such as: museums, restaurants and transports' lines.

POI; moreover, multimedia contents can be completed by manual annotations and comments. If the AR application allows the user to share POIs and annotations, it is said a social application. In this case, the impact of AR technologies might be magnified by the Net as users generally ask for a direct link between applications and their own profiles on social networks. Sometimes, AR applications for tourism and entertainment have a strong relationship. In this second type of AR applications, a tour is organized as a multi-level game and users have to solve mysteries and answer questions related to the tour itself; users receive information about the next stage of the tour only when they are able to complete the current level. This approach provides the so called space gamification. The third type of AR application for tourism is related to the concept of fictionalization. In this case, the tourist experience is augmented referring to very special places such as film sets or locations described in literature: AR can improve and enhance the fictionalized landscape visits. A main issue affects the spread of AR for tourism: the lack of interoperability among applications; this impacts both on developers and content aggregators. If this problem

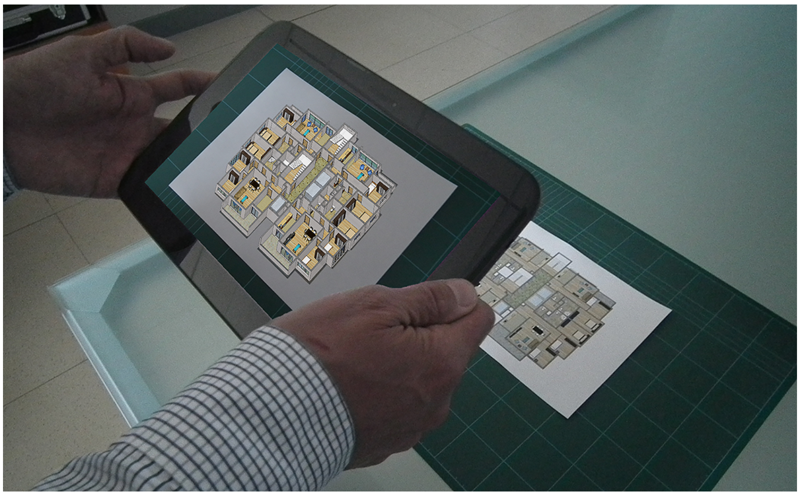

Fig. 1.7 An architecture application, which allows the user to visualize a 3D model of the apartment's blueprint framed by the camera.

is shared by all application domains of augmented reality, tourism is the one more affected because of the huge amount of information gatherable.

1.5.6 Architecture and Construction

AR technologies find an important application field in architecture and construction tasks, which are usually classified under the category 'architectural design and urban planning'. Architects and designers deeply stress issues related to spatial communication: they would intuitively convey information about appearance, scale and features of a proposed project. Unfortunately, neither scale models nor virtual models completely tackle all the needs of spatial communication. AR can be a valid support and several types of application are well known:

1. The first and simplest type aims to display buildings, from their early designs to final constructions, by representing them as virtual objects into the real world. These applications allow architects to plan and evaluate in advance the impact of a new construction (urban planning); moreover, this approach enables the so called walking tours, thus allowing users to move in a real scenario and observing a virtual building as if it were real. Figure 1.7 shows an example of this type of application, with a 3D model overlapped to its blueprint through AR.

2. AR applications can also enable a sort of x-ray view. Hidden features such as structural supports, pipes and electric cables can be overlapped to a building. In this case, it is very important the integration of the AR application with BIM (Building Information Modeling) data.

3. Collaborative design. The final goal of architects and designers is to obtain a seamless transition between individual work with CAD tools and collaborative work. Collaborative work can strongly reduce design times and improve the interaction between designers and customers. Magic meetings refer to augmented/mixed spaces where people can cooperate to model, analyze and assess CAD projects. An example of magic meeting is MR^2, a mixed reality meeting room [52].

4. As presented in section 1.5.2, AR can be used for maintenance tasks; in this case, special purpose applications are tailored to support building maintenance.

At the present time, very few architects and designers use AR, and the latter is mostly used to enhance presentations and marketing. This is due to a lack of integration of AR technologies in the design workflow. Although this problem has been investigated since the last decade [53], a lack of standards prevents a real integration of AR in CAD software.

1.5.7 Cultural Heritage and Museum Visits

AR applications for tourism and cultural heritage share a lot of requirements and characteristics; on the other hand, some distinguishing features can be identified. First of all, it is important to define the concept of cultural heritage attraction. It can be a monument, a ruin, a battlefield and everything can assume a cultural value. At the present time, videos and virtual tours are the main supports to promote cultural heritage; printed info and scale models can be also available for on site visitors. Augmented reality can strongly enhance this support by providing the users with multimedia contents ranging from the temporal reconstruction of a site to a virtual reconstruction of a battle. This kind of applications can be categorized as an augmented guide (see for instance [54]). Figure 1.8 shows an example of an augmented guide that provides textual, audio and video information to the user,

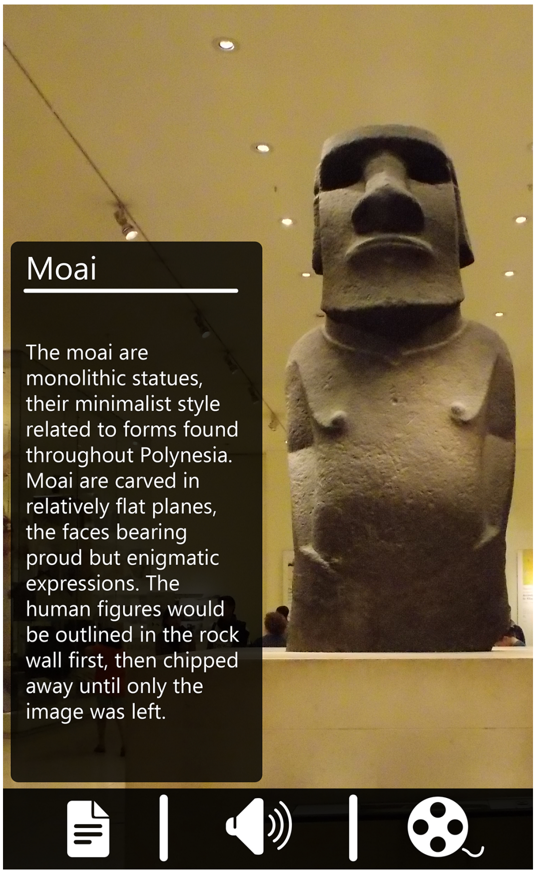

Fig. 1.8 A multimedia guide for museums that provides textual, audio and video information for the work of art framed by the camera.

depending on the work of art framed by the camera. When cultural heritage gives rise to cultural tourism, AR provides a set of undeniable advantages [55]:

1. Printed info and other unnatural objects placed in a cultural site involve costs and, moreover, alter the site itself; AR applications do not need information boards or other ‘artifacts’ and can be modified/updated more efficiently than traditional info;

2. AR does not limit the amount (and the type) of information for users, whereas info boards have a limited size, thus reducing the information potential;

3. AR applications, generally, enhance the user experience and this can trigger a virtuous circle where costs due to the development of an AR application are balanced by the positive publicity of satisfied users that visited the site;

4. Cultural heritage AR applications are often social, thus allowing users to share their experiences by social networks.

A characteristic feature has to be mentioned for AR applications designed for museum visits. Every type of multimedia guide has to be supported by a navigation mechanism able to determine position and orientation of the user. Basically, a guide has to answer to the following two questions:

1. Where can I find the object for which I see available multimedia content?
2. Where are the information related to a particular artwork that I can see?

Navigation systems based on Wi-Fi, Bluetooth, RFID and infrared technologies can either fail (or they can be strongly limited) when the user is moving in crowded environments or provide a not sufficient precision in the computation of the user's orientation. AR tracking systems can mitigate these problems, thus providing developers an efficient solution for localization in indoor GPS-denied environments. Issues related to AR applications targeted for cultural heritage and art are almost the same as AR applications for tourism.

1.5.8 Teaching, Education and Training

Teachers, educators and trainers are always searching for new technologies able to enhance the learning experience of their students. AR proved to be an effective and efficient tool to improve traditional learning and training paths. AR changes the way users and machines interacts and this can stimulate students to approach the study of course material in a different and more pro-active way. Moreover, AR applications tailored to support teaching and learning can stimulate cooperative and collaborative learning, thus improving the teacher-student and student-student collaboration. In the same way as virtual reality, AR helps instructors to simulate and visualize microscopic or macroscopic scale systems; moreover, dangerous and/or destructive events can be represented. A lot of examples of projects and publications are known in the literature (readers can examine in depth this topic by referring to two survey papers [56] [57]) that provide a clear vision of how AR might improve teaching and training methods. Medicine, Engineering, Architecture, Chemistry, Mathematics and Geometry, Physics, Geography, Astronomy, History,

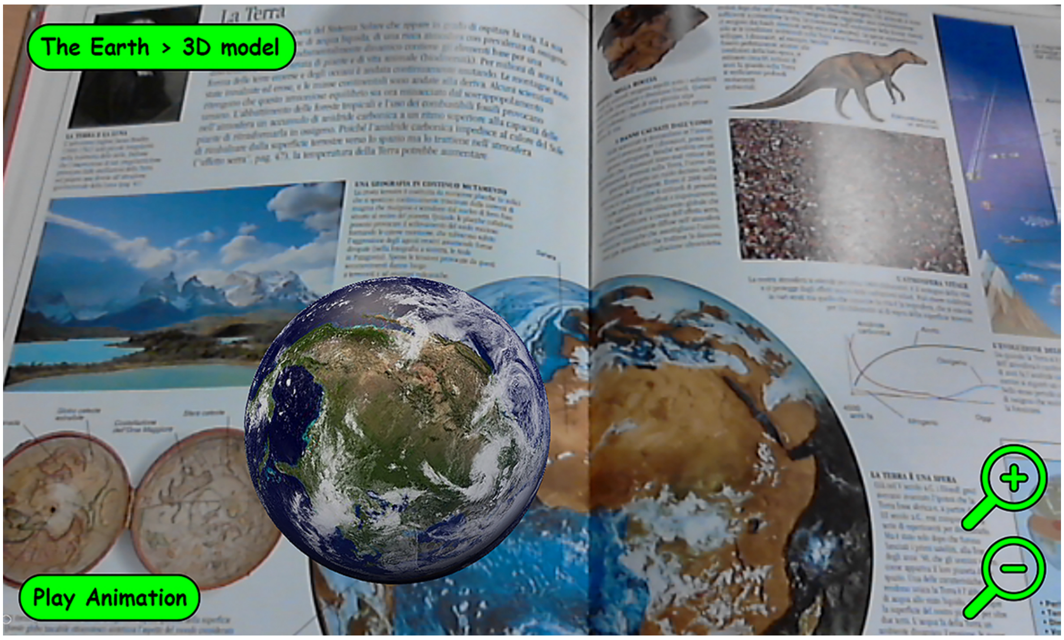

Fig. 1.9 A magic book that provides multimedia content related to the current page framed by the camera, e.g. a 3D model of planet Earth the user can interact with.

Archeology, Music and Art are just some examples of disciplines that might be taught in a different and more exciting way by AR. All domains previously presented as application fields can take advantage from AR technologies for education purposes. Maintenance AR applications can be used to support expert technicians as well as to train beginners and AR games could have an educative value. AR is often used in the discovery-based learning where the recognition of a place (geo-referenced AR applications) or of a person is the starting point to present, for instance, historical events or personalities. On the other hand, the most common use of AR in education is related to interactive course material (often providing 3D visualizations), which allows educators to reduce the gap between real and virtual world. Magic books (also called AR books) are the best example of interactive material: some pages present animations, audio contents and 3D objects, students can interact with. The Magic book allows teachers to implement the so called blended education, that is a hybrid approach that uses two (or more) different teaching technologies (e.g. the traditional book and the augmented reality). Figure 1.9 provides an example of Magic book application: a 3D model of planet Earth is visualized on top of a 2D image of Earth itself depicted on the physical book; the application allows to rotate the model and to change its size to provide further interaction. Despite of all these encouraging

Fig. 1.10 A military application that provides soldiers with additional information on the surrounding area, such as: target position, distance from the base camp and previous events in the area.

aspects, a main issue has constrained and limited the spread of AR in education so far: the difficulty for educators to make quickly deployable AR contents. AR applications and their contents are, at the moment, quite rigid, therefore it is very difficult for teachers to change and adapt the course material to the students' needs.

1.5.9 Military Applications

AR applications for military purposes share a lot of issues with AR-based games and AR-based training systems. The term often used is BARS: Battlefield Augmented Reality System [58]; in other words, AR is used to synthetically create battlefields within the real world, which can be navigated without the limitation common to virtual environments. Usually, training scenarios for soldiers are made by projecting a virtual environment and virtual actors on walls within training facilities. This needs significant infrastructures and it is limited to indoor contexts. AR allows to overcome these constraints, thus allowing soldiers to carry out outdoor training activities where they can physically move through real environments. Another use of AR is ordinary in military applications: the overlapping of information with the environment. For instance, soldiers can receive on their AR vision systems (mainly AR glasses) information about objects in the battlefield and the threat level related to each element of the environment. Figure 1.10 shows an example of military application: different kind of information is overlapped to the real scene to provide additional data of the surrounding area, such as target position, distance from the base camp and previous events in the area. Some characteristic issues are related to the use of AR in military applications. First of all, if AR applications basically aim to track the position of a camera with respect to the real world, also weapons have to be often tracked in AR-based military systems. In first person shooting systems (see for instance [59]), the soldiers' heads have to be tracked as well as the positions and orientations of their weapons. This is necessary to exactly determine what targets have been shot. Moreover, information conveyed to the user has to be carefully managed; in fact, a huge amount of data might be available and an indiscriminate visualization of them will have a negative effect in terms of cognitive overload. This last issue can be partially mitigated by implementing a natural input interface based on speech and/or gesture recognition that enables soldiers to intuitively select and configure information to be overlaid to the battlefield. Another important characteristic of AR-based systems deployed for military applications is the collaboration function. Soldiers have to be able to exchange information (e.g. each soldier should know the positions of the other ones, thus discriminating between them and potential foes), therefore affecting what assets are displayed on the interfaces of the other collaborating users. AR-based systems for military applications should be also able to monitor each soldier's stress level, thus adapting the output of the interface in order to provide the best support.

1.6 Conclusions

This chapter proposes a brief introduction to Augmented Reality. Whereas sections 1.3 and 1.4 describes how Augmented Reality works, sections 1.2 and 1.5 point out the historical importance of AR and its usefulness in many different application domains. The next chapter will focus on AR for maintenance and will present the methodology adopted for the present research.

Chapter 2

Methodology

Part of the work described in this chapter has also been previously published in [3] and [4], [60], [61] and [62]

The widespread adoption of mobile devices is giving everyone access to augmented reality systems, possibly involving a huge number of people in AR-based apps, with a pervasive social impact that cannot be neglected. AR systems are becoming affordable to everyone and, among all the different application domains, it is especially useful to research AR in the maintenance field for many reasons. Maintenance is one of the first field of application identified by Azuma in [10] and it offers all the most common and important challenges that AR can arise, such as:

1. precision: since maintenance deals with machinery, it requires precise alignments of the assets in order to provide a proper aid to the user;

2. environment: maintenance may require the user to work in different light and environmental conditions, so it is necessary to adopt tracking solutions that could adapt to different situations;

3. interaction: the users will often need both their hands to perform maintenance procedures, thus it is necessary to provide effective hands-free interaction solutions;

4. noise: manufacturing environments may present different type of sound interferences, thus it is necessary to research robust interfaces when speech recognition interaction is required.

Moreover, since these problems are common in AR, solutions provided through the research may be adopted also in the other application domains described in section 1.5.

The first step of the adopted methodology was to perform an analysis of the state of the art in order to highlight existing solutions and open problems. The second one was to identify the use cases that could be suitable to address the open problems. After that, user requirements have been collected in order to drive the design step of the architecture.

2.1 Maintenance

The concept of maintenance relates to the adequate care and actions required to ensure equipment functions effectively. This would imply that actions are taken to ensure that equipment does not fall into disrepair. However, the reality for the majority of industry is that equipment is not maintained in proper working order. Instead, machines fail before any actions are taken meaning that the machine is out-of-order and laying dormant until the problem is fixed.

Maintenance is one of the most cost-effective methods for ensuring reliability, safety and energy efficiency [63]. There is therefore an intrinsic relationship that exists between productivity, reliability and maintenance with the primary outputs of effective maintenance leading to reduced operating costs, on-time delivery and consistent product quality [64]. Importantly, poor maintenance strategies also have large environmental impacts with energy losses from steam, water and air leaks, uninsulated lines etc. Improving these practices can therefore lead to substantial energy savings. With increased reliability, productivity, environmental impacts and consistent quality, effective maintenance can therefore have a considerable economic impact for the European economy.

The economic potential of effective maintenance has become increasingly highlighted by Governments around the world, with some Countries going as far as making effective maintenance part of the law. For example, EO13514[65] was brought into effect in the United States in 2009 to direct Federal agencies to further consider energy, water and operational efficiency [66]. Laws such as this emphasize efforts towards sustainable buildings, cost-effective strategies to minimize energy, water and material consumption and promote re-use/recycling techniques. Over the last 30 years, as more emphasis has been placed on maintenance, different approaches to how maintenance can be performed have been developed. The following sections will briefly outline some of these approaches.

2.1.1 Reactive Maintenance

Reactive maintenance refers to the approach where industries use the machine until it breaks. No actions are taken until the machine encounters problems. Sullivan et al. [66] suggest that this approach is still one of the most prominent maintenance procedures. It could be considered that by adopting a reactive maintenance approach, we will not waste labor hours and money until a machine breaks. The period when the machine is working could be considered as money-saving since money is not 'wasted' with maintenance procedures until issues arise.

Indeed, it may in fact be the case that more money is being wasted than if a different maintenance approach were adopted. That is, whilst the equipment appears to be working fine, the life of the equipment is being shortened resulting in more frequent part/tool replacements. Furthermore, when one issue does arise, it may have a secondary impact on other parts causing increased maintenance costs to address not only the primary issue but also the knock-on effects. These costs could be avoided with alternative maintenance strategies. It also seems fair to suggest that a reactive approach may also lead to failures that are more complicated and potentially more time-intensive to fix. If an alternative maintenance approach were used, then small issues could be addressed immediately before they develop into more complicated issues. A knock-on effect of complicated issues is that they will cause the machine to be off-line for longer and therefore create greater productivity loses.

2.1.2 Preventative Maintenance

A preventative maintenance approach relies on a series of tasks performed regularly to either extend the life of a machine/part or detect critical wear and predict failure

[67]. Such techniques (e.g lubrication and filter changes) therefore detect, preclude or mitigate degradation of machine tools by controlling degradation to ensure it remains at an acceptable level.

Over recent years, a wealth of research has focused on the impact of preventative maintenance and its associated costs. In particular, much of this research has used mathematical modelling to investigate the use of preventative maintenance for a range of tooling parts. For example, Chelbi and Ait-Kadi [68] used a mathematical model for joint strategy of stock production and preventative maintenance for a randomly failing production unit where preventative maintenance durations are random. Juang and Anderson [69] adopted a Bayesian theoretical approach to quantify an optimal adaptive preventative maintenance policy with minimum repair. Sullivan et al. [66] summarize that the US navy utilized preventative maintenance strategies to enhance reliability of their vessels. They found that by sparing the resources to conduct maintenance activities as planned by equipment designers, the equipment life was extended and its reliability increased. Interestingly, money was saved when compared to a reactive maintenance program. Sullivan et al. [66] suggest that these savings can be as much as 18%. This approach may not be sufficient enough to prevent catastrophic failures but it will decrease the number of failures. Reducing these failures leads to economic gains.

2.1.3 Predictive Maintenance

Predictive maintenance aims to detect the onset of degradation and failure in an attempt to eliminate/mitigate these issues prior to significant deterioration. The main difference between preventative and predictive maintenance approaches is that predictive maintenance is based on the actual condition of the machine. In contrast, preventative maintenance acts on a planned schedule. For example, changes of filters and lubricant will happen every x number of hours of use etc. In this way, the lubricant and filter are not assessed to see if they need changing but are changed as a set (predetermined) time has passed. Using a predictive approach, the lubricant or filter would be periodically assessed so that the condition and properties could be determined. If no issues are apparent then the change may not occur until a later date, thus extending the life of the lubricant or filter in this instance.

The main advantage of predictive maintenance is that catastrophic failures should (hopefully) be avoided. More pertinently, as the maintenance is planned, overtime costs should be avoided as required parts/tools can be ordered ahead of time before they are required. That is, if a part looks to be starting to wear, a new part can be ordered before the part needs essential replacement. As predictive maintenance should lead to an optimized machine/tool, there will be evidence of energy savings and increased reliability. Sullivan et al. [66] outline that a predictive maintenance program could save 12% over a preventative maintenance program alone. It was outlined previously that preventative maintenance may save money when compared to reactive maintenance so it is clear that predictive maintenance will save considerably more than reactive approaches.

It is important to note, that predictive maintenance can be difficult for everyone to implement. Although they save money in the long-run, the initial costs are quite considerable, meaning that some companies, in particular SMEs may struggle. To implement predictive maintenance, it is essential to invest in the appropriate equipment, which can be costly in it-self. In addition, staff needs to be trained on how to adopt the predictive maintenance technologies.

2.1.4 Reliability Centered Maintenance

Reliability centered maintenance originated from the aircraft industry but has now been adopted with considerable success across a wide-variety of industries [70]. Such an approach assesses the equipment functions and how/why they can fail and bases maintenance actions on safety and economic priorities. This approach therefore deals with some key issues that are ignored by the other approaches outlined previously. It prioritizes different aspects and therefore does not recognize all the equipment as equally important. In addition, this approach acknowledges that some parts have a higher probability of failure than others. It can almost be seen as a pragmatic approach in that it acknowledges that companies do not have unlimited labor-force hours or financial resources so these need to prioritized and used where they can have the biggest impact [66].

A reliability centered maintenance approach can therefore increase both reliability and cost-effectiveness. Due to the manner in which this approach prioritizes actions and equipments, it incorporates many of other approaches discussed previously. For example, for high priority processes and equipments it relies heavily on predictive maintenance but for less important or cheaper equipment a reactive maintenance approach may be adopted.

2.2 Augmented Reality in Maintenance

The previous section illustrated the range of traditional maintenance strategies that can be adopted. However, it is also important to consider different techniques and tools that may be used to aid the maintenance process. One such technique that is becoming particularly relevant in recent times is augmented reality.

Most of academic and advertising photographs of conventional head mounted displays (not AR) try to advise that these products will turn the multiple task performance in maintenance jobs into a single task performance, but unwittingly, these imageries suggest just the opposite (e.g. Fig. 2.1). These photographs often show a wearable video display that creates superimposed graphics without any kind of registration. The images depict the wearer in what looks like a trance-like state, maybe holding a tool or touching a work piece, gazing at the display but appearing to be completely detached from the work task. This is precisely the state expectable if the user must pay attention to a display that is independent of the equipment. But a user of a head mounted display hosting AR is not expected to appear detached from the work task, simply because presentation and retrieval of information is integrated with views of the work piece.

Augmented reality, in fact, can trigger the appearance of virtual elements by the user's view of real equipments and structures. For example, in one application study, the act of looking at an electric system allows workers to see wiring harness assembly instructions. This dynamic differs a lot from technicians' typical interaction with information about the equipment. Normally, the information is detached from the apparatus, except in the case of control panels and where lighting, frequency of use, and the size of parts allow labels or tags to be positioned on them. The worker searches a medium for information, such as paper manual or laptop computer, often in the form of an annotated drawing or picture. AR can spare the worker the search: the act of looking or directing a camera at the equipment is sufficient to produce information.

Fig. 2.1 Example of head-mounted display.

On 2004, Navab [71] identified a set of possible high-impact applications for industrial AR. Service and maintenance played again a key role. Moreover, the author outlined how three main issues had to be addressed by 'useful' AR-based applications: reliability, user friendliness and scalability beyond simple prototypes. Maintenance, repair and assembly are still identified as strategic application fields, since the reduction of associated costs represents a key goal in many domains. Indeed, cost changes would depend on the particular application scenario. For instance, maintenance and repair costs make up just 4% of the ownership costs of a car [72]. However, in the aircraft domain, maintenance costs can reach 80% of the entire product life-cycle (from design to disposal) [73], and have been expected to increase overall to $54 billion in 2015 [74]. Hence, any technological advancement is carefully considered in order to take the opportunity to reduce these costs. Even though Interactive Electronic Technical Manuals (IETMs) are frequently used to replace paper-based instructions [75], they are not completely part of the interaction between the technician and the equipment to be maintained. Therefore, both the overall time of the repair/assembly task and the cognitive load on the maintainer might increase [76] [33]. This could also affect the number of errors introduced in the maintenance process.

2.2.1 Maximize Human Performance

It may seem obvious that eliminating the search for instruction would benefit a task. However, there is a long and accepted tradition that the recovery of information in some detached medium (usually paper) is a normal part of tasks. The two types of activities are so connected that it is easy to ignore the impact of informational activities on work task performance. For example, aircraft maintenance evokes images of repair actions on actual hardware, but an airline expert reported that 45% of every technicians' time is actually spent on finding and reading procedural and related information [77]. Of course, observations can reveal that tasks involve different activities. From a cognitive standpoint, the skills and abilities invoked for these two requirements are very different, and they are often invoked sequentially.

Further evidence that searching for information and other activities related to instructions is different from work piece activities comes from Douglas Towne [78]. He measured the time for two types of behavior in isolated tasks: time for actual manipulation of devices and instruments (manual time) and time not engaged with devices or instruments (cognitive time). He found that cognitive time accounted for about 50% of total task time [78]. More importantly, cognitive time was independent of manual time, meaning that individual technicians differed in how much time they devoted to cognitive/informational tasks, but differed little in how much time they devoted to manual jobs.

The independence between manual time and cognitive time is often regarded as evidence that it is possible to adopt different techniques and approaches for enhancing both types of task, in order to improve the overall maintainers performances. Moreover, experience and insights reveal that manual and cognitive activities generally happen sequentially rather than concurrently, unless they are highly practiced. Therefore, it is possible to conclude that if cognitive activities had been reduced for the fault isolation technicians, their total task time would have been lowered. This is just one of the benefits envisioned for AR in maintenance tasks.

Finally, the differences between information and work piece activities indicate that when they must both be accomplished to reach some end, they together qualify as a multiple task performance. Rogers and Monsell have shown that it is easier to keep doing alternate versions of the same task than it is to switch between different tasks, indicating the presence of some overhead chore, such as retrieving "rules" associated with each task [79]. AR is expected to lower the frequency of transferring between information and work tasks and therefore to reduce the time and energy demanded in this repetitive exchanging.

2.2.2 Reduce Potential Error

When tasks are repeated frequently, maintenance operators can become experts in those tasks through the combination of low performance variability and the so called “over-learning”. The likelihood of errors is often a function of the interaction of individual factors such as a worker’s expertise and situational factors such as the task environment. For example, Miller and Swain studied that novices and experts are equally likely to err in performance under low stress, but novices are more likely to err under high stress.

AR can endow novices with some of the advantages enjoyed by experts, such as an efficient retrieval of information from memory, irrespective of the situation. AR can provide this expert status in two ways. The first is simply the basic effect of AR: the triggering of information with little user effort. Maintenance experience is filled with evidence that people favor information that is easy to access and tend to use more salient data in decision-making. Conversely, technicians resist the effort involved in accessing remote or distant information. In a related situation, information from head-up displays (HUDs) in aircrafts has a much lower “access cost” and is therefore more likely to be scanned than is the same information from instrument panel displays just a few inches away.

The other way AR reduces error is by speeding the user’s transition from “information novice” to true, unaided “information expert”. This transition is facilitated by dynamics of AR that complement human associative information processing and memory. First, virtual objects have locations, and second, virtual objects can be associated with real-world features.

2.2.3 Enhance Motivation

Multimedia can generate a rich sensory experience that not only delivers information, but “appears to increase the motivation and interest of the reader or viewer”, as Chignell and Waterworth [80] stated. A thoughtful question is whether AR media could be designed to engage a user in a task with the help of some effects that only multimedia can produce. Can the media, apart from its informational value, be compelling enough to become the technician's preferred tool? Could the media increase observance of correct procedures? Kanki and Veinott found that in the Aviation Safety Reporting System (a NASA-conducted method for airline personnel to report problems anonymously), 60% of maintenance-related reports concerned procedural errors, and many of these errors were due to negligence [81]. Also, negligence and cockiness played a big part in a company's high injury rate [82]. The point these examples illustrate is that some deficiencies in maintenance are not due to lack of information.

AR is believed to be a candidate to produce better adherence to correct procedures for the virtue of increasing motivation. Today, the technology creates a basic multimedia experience for the user, and in time, virtual objects in AR will take on more roles, be more smoothly integrated into the real-world scene, and offer greater interactivity. Because the computer-generated overlay is under program control, there are many choices to increase the interactive multimedia aspects of AR. A virtual object, for example, can be a symbol, video/audio clip, menu, CAD solid model, synthetic instrument face, communication window, input button, agent icon, or any other object presentable by computer. The options are nearly limitless, and are set up by the planning and authoring stages in an AR system development. Some of the potential uses of this control are the following:

1. design eye-catching objects to influence the user's focus of attention (for example cautions or warnings);
2. design objects to be adjustable and customizable (for example to appear more or less distant from the viewer, or to provide text in different languages);
3. make objects dependent on operating conditions (for example higher contrast call-outs in bright viewing conditions);
4. enhance user's ability to organize elements into functional sets (for example overlay color coding on pipe runs);
5. allow users to invoke a virtual "copy and paste" to keep information accessible (for example to reduce the need for reinstatement searches).

2.2.4 Research Topics

AR-based Documentations

Benefits of AR-based documentation for maintenance are well depicted in [45] and [83], where it is outlined how AR technology can reduce costs up to 25% and improve performances up to 30%. Moreover, although over the past decades AR-based applications for maintenance were used only by technical specialists, current solutions promise to profoundly change the way end-users will perform many of their daily tasks. For instance, traditional paper-based owner's manuals for ordinary car maintenance, furniture building instructions and installation manuals of electrical appliances could be soon replaced by AR-based applications on mobile devices. The number of people possibly involved in using AR-based applications is potentially huge and the social impact cannot be neglected.

AR-based Training Systems

Augmented reality has been deeply investigated and used in order to improve traditional learning and training paths. The possibility to create enhanced user-machine and user-user interactions by AR technologies has been the basic motivation for a lot of researchers in designing and developing AR-based systems to support teaching and learning. Moreover, AR can be also an incentive for students, thus motivating them to analyze more in the detail course materials (see section 1.5.8). AR can help instructors to simulate dangerous or destructive events as well as can help learners both in visualizing microscopic/macroscopic scale systems and in effectively collaborating with teachers and other students.

Interfaces for AR-based Systems

Today, affordable, head-mounted and see-through displays are available for many consumer-oriented purposes, like gaming, tourism, education, etc. [84–86]. While opening up enormous possibilities, the technological evolution also brought to the fore significant challenges regarding human-machine interaction. In fact, many paradigms used in traditional settings are no more applicable to mobile and wearable devices. Hence, researchers are experimenting with different modalities, often combining some of them into mixed interfaces and performing comprehensive user studies to determine their applicability in relevant scenarios [87].

Multi-modal interfaces proved their viability in supporting complex interaction tasks [17]. It is well known that, in dealing with these tasks, if the design is accurate, it can benefit from the advantage of a specific interaction modality, besides simultaneously limiting the impact of its drawbacks. For instance, mouse and gestural commands can be effectively exploited for direct object manipulation (e.g., for positioning virtual elements using 2D or 3D coordinates), while voice can be used at the same time for descriptive tasks (e.g., for choosing what to work with). Thus, for instance, Irawati et al. used speech and hand gestures for interacting with virtual furniture in a wearable-based AR application in [88]. A similar approach was adopted by Koelsch et al., who used voice commands in combination with a hand-held trackball to interact with AR contents during maintenance and emergency rescue tasks [89].

On the other hand, unimodal interfaces are generally exploited in simpler interaction scenarios. Common unimodal approaches for interacting with large-scale VR solutions exploit speech, wands, laser pointers, as well as hand and body gestures, whereas AR applications on mobile devices are often managed by means of native touch controls and voice commands [90], [91]. The foregoing modalities are not always applicable to setups based on wearable devices. In fact, some situations require only hands-free operations, e.g., when executing maintenance, medical or other hazardous tasks [75] or improving service productivity and efficiency [92]. In these situations, generally speech interaction is considered.

2.3 Related Works

Early examples of conveying maintenance instructions to technicians by using AR-based systems, which can be dated back to the 1990s, are described in [85] and [86]. AR-systems are usually classified based on different parameters, including tracking technology, human-system interaction method, data management strategy, etc. A rather common categorization considers the visualization device. Predictably, a key role is played by wearable technologies [87]. In particular, since from the early experiments by Feiner et al. [17] with a head-worn AR prototype designed to support end-users in performing simple maintenance procedures on a laser printer, a number of head mounted display (HMD)-based solutions were developed. For instance, in the Etälä project [93], tele-assistance and AR were exploited to establish a communication channel between maintainers and remote experts (in a so-called tele-maintenance framework [94]). Experts could navigate a virtual model of a real object and share useful information to support maintenance and repair operations. The same approach was adopted in [90], where VR and AR were used to remotely support technicians as well as trainees. The German Federal Ministry of Education and Research (BMBF) sponsored the ARVIKA project [91], which was aimed to the design and implementation of a head-worn AR-based user-centered tool to support the development, production, and servicing of complex technical products and systems. STARMATE [92] was one the first examples of multimodal augmentation, where virtual objects, textual hints and audio messages were used to guide and support the maintenance of mechanical parts.

Collaborative frameworks have been also proposed. For instance, AR-technologies are used in [95] to train technicians in assembly tasks of complex systems such as aircraft engines. Several studies analyzed the potential of such systems and their impact on training and maintenance processes, by considering different application scenarios. As a matter of example, the ARMAR project [76], [96], [97] considered advantages and drawbacks of various hardware and software solutions for maintenance job aiding in the military field. Ke et al. developed a prototype to maintain PCs [98]. The ARVIKA project addressed the automotive and aircraft industries. Results produced by ARVIKA represented the input for the ARTEAS project [99], which explored the use of AR technologies in more general contexts. Schwald and Laval focused on the usage of AR solutions for maintenance operations in generic industrial scenarios [100]. In [101] and, more recently, in [102], AR was used to simulate and validate the programs of Computer Numerical Control (CNC) machines. Efficiency of AR in the manufacture industry was specifically measured in [103], where performances of people involved in an assembly tasks have been measured in terms of time saved and errors avoided due to either wrong tool selection or wrong part positioning.

In some works, like in [104] and [105], AR is referred to as mobile to denote the possibility for the users to move, typically in an industrial site, in order to find objects to be maintained. For instance, the goal of the PLAMOS project [105] was to support owners and operators in plant maintenance and repair. Specifically, a marker-based solution was used to easily identify industrial facilities throughout the

(a) (b)

Fig. 2.2 Handheld Augmented Reality for maintenance operations: a) real object under maintenance framed by a smartphone, b) close look on the HAR application graphics interface.

plant. Markers were used in a similar way in [106] to support factory planning in the automotive industry. In [107], an AR-based Operations and Maintenance Fieldwork Facilitator (AROMA-FF) was proposed, with the goal to reduce the time needed for the end-users to locate target objects.

Portable devices, such as smartphones and tablets, support a different AR paradigm. In fact, if HMDs enable see-through visualization, mobile devices implement the so-called hand-held display paradigm (see Section 1.3.3). As reported in [108] and [109], Handheld Augmented Reality (HAR) has been fueled by the spread of personal devices and a lot of solutions are already available. However, HAR applications in the maintenance domain are still quite scarce. An example is reported in Figure 2.2, where the use of a smartphone for repairing a netbook is illustrated. The technician frames the object under maintenance using a smartphone (Figure 2.2a). When the object is recognized, some virtual cues, e.g., a text message and a computer-generated animation, are aligned and displayed on the screen (Figure 2.2b). A new configuration of the object has to be recognized at each step of the procedure, which evolves according to a predefined state diagram. By considering Fig 2.2 it is immediately clear that at least one hand of the maintainer is used to hold the HAR device. Hence, is not possible to have both hands free to perform the task. Moreover, the observation of the real object through the device camera might involve safety issues. Despite possible limitations, some examples of HAR-based maintenance are known in the literature and a Fiatech's webinar presented benefits coming from the use of AR applications on mobile devices [110]. Thus, for instance, Kahn et al. proposed a HAR solution able to support the overall life-cycle of construction and facility management [111]. AR-based training systems for maintenance applications using mobile devices are illustrated in [112], [113] and [114]. Kim and Moon [115] focused on car maintenance and introduced a new and potentially high-impact field of HAR, i.e., the replacement of handbooks and paper-based instructions.

Although many applications of AR could get a higher social value, solutions of AR-based maintenance tailored to end-users, i.e., not for specialists, are quickly growing as they promise to have a significant influence on everyday life. In fact, any dematerialization process is expected to produce a social impact [116] and, sometimes, the traditional interaction procedures can be even toppled. Car routine maintenance is one of the fields that is doomed to be profoundly changed by the introduction of AR-based solutions. Some car makers now provide their users the owner's manual also as an application for mobile devices. For instance, it is already possible to recognize more than 300 individual components of the Audi A3 both on the instrument panel and under the hood. In this way, relevant how-to information and virtual overlays of maintenance procedures can be conveyed to the user directly onto his or her personal device. For example, after framing the engine with the device's camera, the AR application could provide an animated overlay of virtual objects, with instructions on how to locate the engine coolant and refill it to the appropriate level [117].

Some assembly tasks could also take advantage of AR technology [118–120]. For instance, assembly of pretty simple objects, such as furniture items, could be effectively supported by AR solutions. Several applications for mobile devices already provide end-users tools for a virtual preview of customizable furniture in the environment, thus implementing an enhanced digital catalogue [121]. Other examples are represented by solutions developed by Mitsubishi Electric [122] and NGRAIN 3D[123]. In the first case, installation and maintenance processes of heating and cooling products are explained by an AR application. In the second case, an application for training on the maintenance of industrial parts is provided. Some companies also quantified benefits obtained by adopting AR. For instance, a toymaker (Lego) provided users a tool for enhancing the assembly experience and observed that sales increased by 15% after the introduction of the tool.

Examples above clearly show the potential impact of AR on daily activities in the maintenance, repair and assembly domains. Nonetheless, as it will be detailed in section 2.4, there are still problems that need to be addressed.

2.3.1 AR-based Training Systems

Several fields and disciplines benefited of AR for education purposes; the following list is not exhaustive (a survey is out of the scope of this thesis) but it is aimed to provide readers a picture of the impact AR can have on everyday life. Several applications have been proposed for the education of: medicine (e.g., [124, 125, 44]), engineering (e.g., [126, 127]), architecture and interior design (e.g., [128, 129]), chemistry (e.g., [130, 131]), mathematics and geometry (e.g., [132, 133]), physics (e.g., [134, 135]), geography and astronomy (e.g., [136, 137]), history and archeology (e.g., [138, 139]), art and music (e.g., [140, 141]).

A lot of works have been also proposed in the more specific field of training for maintenance. The idea to train and support technicians by conveying computer-generated instructions can be dated back to early 1990s (the reader can refer to two surveys, [86] and [85]). In particular, Feiner et al. [17] showed potentialities of AR-based applications for repair and assembly tasks by supporting maintenance procedures of a laser printer. AR technologies are now used for training and support technicians in a large number of application domains: aerospace [142, 143], automotive [144, 145], industrial plants [146, 147] and so on. Benefits of AR to support maintenance, repair and assembly tasks are well investigated and presented in [45].

As AR technologies allow researchers to develop user interfaces able to reduce the gap between real and virtual objects, a lot of works are known in the literature about AR books (e.g., [148, 149]). AR books allow to provide students interactive material and 3D visualizations, thus implementing the so called blended education (a term used to identify a hybrid approach that uses different types of training technologies). At the same way as AR books, AR games are a type of education that allows teachers to use a highly visual and interactive form of learning: Human Pacman [150], AR2 Hockey [151] and ARQuake [46] are just the first examples of a new frontier of AR-based education (see Section 1.3).

Despite of the last decade, when AR systems were mainly based on special purpose hardware, the evolution of mobile (personal) devices allows to replace the see-through AR-interfaces by means of hand-held AR-interfaces; as the best part of mobile devices (smartphones and tablets) is endowed with a GPS, the discovered-based learning [152] is growing up. Discovered-based learning is not only based on geo-localization (often used to teach history or geography) but also on face recognition (to provide information about a person) and, more in general, on object recognition. All these examples show different forms of teaching/education by using augmented reality.

2.3.2 Interfaces for AR-based Systems

Virtual and augmented reality applications use several input modalities. Traditional input devices such as keyboard, mouse and joystick have been soon replaced by more sophisticated and natural interaction ways. For instance, touch and multitouch surfaces are used to navigate virtual worlds [153] as well as to remotely manipulate parts of a robot by an augmented interface [154]. Hands and body gestures can be an efficient and intuitive way to convey inputs; wearable [155–157] as well as "desktop" solutions [158–160] based on commercial tracking devices such as the Leap Motion or the Microsoft Kinect are able to use gestures and poses to navigate and, more in general, to interact with virtual and augmented environments. Tangible interfaces are another well known input solution for VR & AR applications; tangible interfaces are well suited for manipulation tasks [161, 148, 162]. When any form of hands-based interaction cannot be used, an alternative interaction technique is necessary; in this case, gaze and speech are usually considered. For instance, gaze interaction is considered in [163] and [164], whereas vocal commands are used in [60], [165] and [146]. Also, brain interfaces have been profitably applied to VR & AR worlds (see for instance [166] and [167]).

Speech Interfaces

Speech interfaces can be exploited to design two broad classes of applications, namely dictation and command & control [168]. Applications relevant to the current work belong to the second class. The spectrum on command & control interfaces ranges from simple prompt and response interfaces to full sentence conversational agents, which have been experimented in a wide range of application scenarios encompassing air traffic control [169], medical practice [170], military operations [171], etc.

Independent of the targeted application, using a speech interface has to address several challenges, such as converting the acoustic voice signal into a sequence of words and translating voice strings into a format that is understandable to machines. Besides such technological challenges, which are beyond the scope of this work, a key conceptual issue is cited as one of the major causes for poor user experience of those who use speech interfaces: “How to make the users aware of what they can say?” [146]. This problem is the same as the problem one faces in passing commands from command- line to graphic interfaces. In command-line interfaces, the boundaries of what can and what cannot be done, i.e., the functionalities available become evident only on knowing the overall set of issuable commands [168]. Hence, graphic interfaces were designed with the goal of visually displaying such functionalities to the user through evocative graphic signs.

In many speech-only applications, like automatic interactive voice response systems, the above problem is tackled by helping the users in the interaction with available commands, using the so-called vocal prompts [172]. But, this approach has a limitation: the speech output is slow, and the users might, therefore, find it difficult to remember all the available commands [173]. When a speech interface is used in combination with a display, cues about available commands may, actually should, be provided. The cues can leverage on the user’s visual memory, thus reducing the cognitive load and limiting, at the same time, interference with the verbal composition processes. This principle is ignored in what is often referred to as a “voice command bar” [174], where text labels are used to display valid commands. The bar acts both as a reminder and as an education tool, making the learning curve smoother and the commands easier to remember. Effectiveness of this paradigm, first introduced by Kurzweil as Applied Intelligence and known as “say what you see”, has been demonstrated in different scenarios [175, 176].

Building on the importance of providing a visual representation for voice commands, Danis et al. argue that, for creating a successful speech interface, the commands must be made mentally available in the easiest way, not only by always displaying the words on the screen, but also by showing to the user more examples of what can be said [177]. This approach has been successfully adopted also in the context of AR [146].

As the number of commands increases, implementing a text-based voice command bar becomes harder. Hence, text labels have to be replaced by icons [178], assuming that each icon is suggestive of one or more voice commands. However, even after using visual cues, mapping the issued commands to the expected functionalities is not a straightforward task. With this perspective, Shriver and Rosenfeld created the so-called universal speech interface by statistically analyzing users' preferences of the commands to be used for activating common functionalities [179].

2.4 Open Problems

The deployment of an AR application may rise up a lot of challenges but clearly the one that plays a key role in most AR-based applications is the precise alignment of the graphical assets in relation to the real world. For example, when dealing with AR systems for medical purpose, virtual data are usually overlaid to the patient's body and it is mandatory to preserve the exact position and orientation of the data. To address this issue, the main problem is to precisely compute the user's head position and tilt with respect to the real world: in an AR application, the tracking system is the component responsible of this calculation, which provides the application with the position and orientation of the user into the real world.

This section aims to present technical problems pertaining the design and development of AR-based applications for supporting maintenance, repair and assembly tasks (other issues, e.g., about privacy and security, are out of the scope of the present work). Indeed, the evolution reached by existing AR Software Development Kits (SDKs) and libraries such as Metaio [180], Vuforia [181], ARtoolkit [182], etc. is so high that designers and developers can focus today on developing application logic and contents. However, it has been shown that there are two main issues, specifically pertaining the training required to setup the tracking mechanism and the reconfigurability of the overall system, that still require a special extra attention.

2.4.1 Pose Tracking

The first issue is related to the capability of AR applications to recognize objects in the real world and track their pose. Approaches based on the use of artificial elements, like markers, displaced in the environment (e.g., sticked to the objects to be tracked) are very robust and work fine. Training the system to work with such elements is also quite easy, since they are designed to be clearly distinguishable. However, it is not always possible to add such information. Hence, a tracking based on natural features, i.e., on object characteristics that are inherently available, is often desirable. In this case, the tracking system can be trained basically in two ways: by images, or by 3D CAD models.

With the first alternative, training is performed by collecting several pictures (referred to as training images) of the real object from different viewpoints. Training images are processed offline in order to identify and extract a certain number of so-called image descriptors. During operation, the above processing is repeated online on images gathered by the user device's camera. When a match between offline and online-computed descriptors is found (a threshold is generally set to define recognition precision and robustness), the tracking step can be executed, thus correctly aligning virtual objects to the real ones. Unfortunately, this approach is strongly dependent on environmental conditions as well as on possible changes in the surface of the real object with respect to what is actually pictured in the training images. Different lighting, reflections, shadows, dust, dirt, rust, etc. may compromise both the recognition and the tracking of the object. A possible solution could consist in replacing training images with photorealistic renderings. Starting from 3D CAD models of real objects, it could be possible to simulate and render different environmental conditions as well as different settings.

However, a more effective alternative is to directly train the system using the 3D CAD models. In this case, the model is used together with a line representation of the model itself (a kind of silhouette-like drawing), which allows the tracking system both to recognize the object and to correctly align the camera to it. Figure 2.3 shows a tracking system trained by 3D CAD models that is able to recognize an inkjet printer. A flat shaded and transparent representation of the model helps the user to align the camera with the real object. The user can adjust the number of polygons of the model to control tracking robustness and precision. The advantage of this second approach is that it is independent both of environmental conditions and of other influences that might modify object surface appearance. The drawbacks are related to the overhead due to the modeling phase of polygonal and line representations, which can be a serious constraint for the development of "home-made" applications.

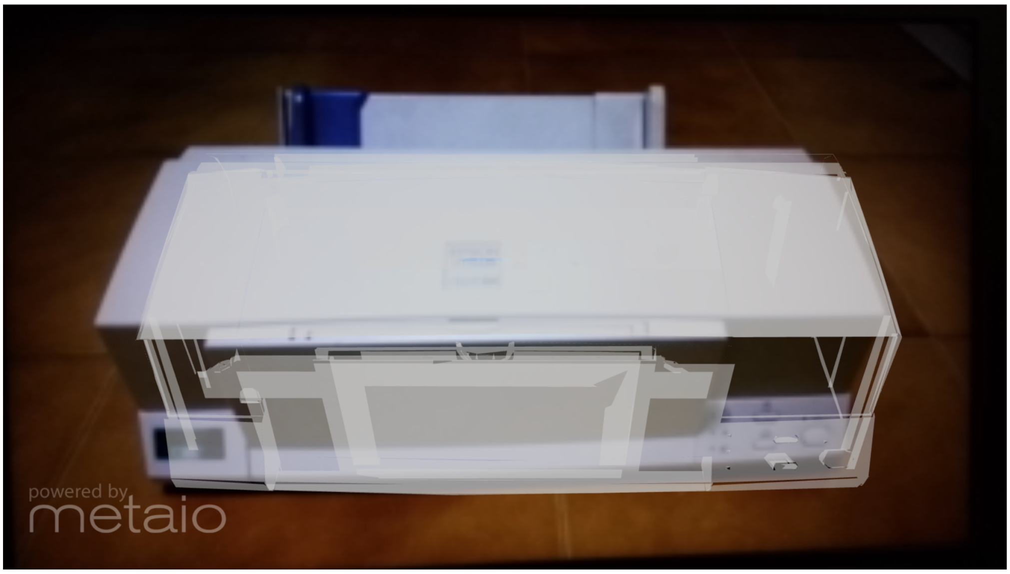

Fig. 2.3 A representation of the 3D model helps the user to correctly align the device's camera with the real object.

2.4.2 Reconfigurability

The second main issue is related to system reconfigurability. AR maintenance applications are usually designed to support a set of procedures that consist of a fixed number of well known steps. As discussed in section 3.3.4, the design of a state diagram is at the basis of any procedure. Each state corresponds to a well defined step of the procedure and system behavior needs to be clearly specified by means of state transitions. However, a dynamic reconfiguration of the state diagram that might be required, e.g., to deal with unexpected situations not contemplated in the existing procedure, is, in general, a very complex task that is hard to be performed at runtime.

2.4.3 Interfaces Design

Despite such huge potential, a lot remains to be done to let users properly interact with virtual and augmented contents in all possible usage scenarios. Interesting reviews of existing user interfaces (UIs) for VR and AR are available [183, 184], where interfaces are generally categorized into the following main groups: text-based, tangible, haptic and tactile, gaze- based, visual (e.g., relying upon gesture recognition), and aural (e.g., using speech recognition). It is worth noting that such reviews did not consider other UIs that are currently being experimented, e.g., the UIs based on brain or other biological signals.

The extreme heterogeneity of envisioned application scenarios does not allow any specific category of UIs to be identified as the dominant one. As a consequence, the last few years have witnessed an incredible proliferation of UIs implemented on ad-hoc basis, which renders reusability of results extremely hard to pursue. Although a number of guidelines have been developed [185, 186], often derived from general human-computer interaction design principles, turnkey solutions for developing UIs comparable to those available for developed UIs, e.g., for creating graphic UIs, are still scarce. A conventional approach is to provide the developers with APIs implementing basic functionalities (e.g., for gesture recognition, eye tracking, etc.) and then leave them with the burden of building the UI.

2.4.4 Robustness of the Interface

Virtual and Augmented reality applications usually provide very sophisticated and efficient interfaces based on different input modes, which are often used together to deploy multimodal user interfaces (MUIs). On the other hand, hands-free tasks (e.g., maintenance and assembly) can benefit neither of solutions based on touch, gestures and poses nor take advantage of haptic-tangible interfaces; moreover, wearable systems might further limit possible input modes. Gaze and speech-based interactions are widespread to tackle potential issues related to hands-free tasks. Although gazing is a very expressive and natural form of human interaction/communication, a special purpose hardware is usually necessary to implement robust interfaces; moreover, gaze tracking devices can prevent the use of other devices such us VR & AR glasses.

Speech-recognition systems are now able to provide extremely high ratios in correctly recognizing vocal commands and verbal communication is one of the most powerful and expressive forms. Unfortunately, performance of speech recognition systems are strongly affected by noise. The environmental noise of an industrial plant as well as the hubbub in extremely populated environments might make speech recognition-based UIs unusable.

2.4.5 Training Systems

All the examples described in section 2.3.1 show different forms of teaching/education by using augmented reality, but they all share a common issue: the difficulty for teachers to create AR contents [187]. For instance, in some AR systems the teaching sequence cannot be changed/adapted; in other words, instructors are not able to (efficiently and easily) accomplish students' needs.

2.4.6 Acceptance of New Technology

Another issue that is worthwhile to be briefly discussed is related to the acceptance of a new technology in a well consolidated process chain. AR could provide unquestionable benefits, but it forces designers and technicians to cope with new tools and manufactures to sometimes rethink their business model related to assistance. This kind of technological change should be carefully managed by deeply involving people in charge to use these new tools and considering suggestions and issues raised by them. This approach has been adopted in the project by involving technicians of Fidia and Wires Engineering in different steps of design and implementation steps of the AR tools.

2.5 The EASE-R^3 Project

The EASE-R^3 project aims to develop an integrated framework for a cost-effective and easy Repair, Renovation and Reuse of machine tools within modern factories. EASE-R^3 will cover the entire lifecycle of the above framework, from the design stage through the operative life. The project, which started on July 2013, is co-funded by the European Commission under the FP7 Framework Program and involves 14 partners from 7 European countries.

Machine tool makers represented in EASE-R^3 expressed the need to provide their customers an efficient assistance. However, arranging customer services in different countries can be very expensive. Thus, technicians often travel from the headquarters to reach the customer's site. Moreover, machine tools are often customized based on customer's needs and cannot be considered mass production objects. Custom objects can require custom procedures, which are very difficult to standardize and define.

In the scenario depicted above, AR solutions are expected to be used for remotely supporting technicians, by means of a tele-assistance approach, in repair and maintenance tasks for broaching, milling and stone cutting machine tools. Despite the vertical domain, the project aims to develop a flexible methodology to allow experts at the headquarters to dynamically reconfigure/modify AR procedures to be implemented by technicians at the customer's site. Among the partners of this project, two companies provided us suitable use cases for our research: FIDIA [188] and WIRES [189].

2.5.1 Use Cases

TMSC Lenses Cleaning

The use case selected for FIDIA is related to the accessory TMSC100, a laser device for tool measurement which measures the length, the diameter and the shape of ten different types of tools. Positioning the TMSC100 on the machine tool at the edge of the working area and executing tool measurement cycles in the intervals between machining cycles are operating conditions that expose the TMS to direct and continuous contact with machining residues (chips, dust, liquids, etc.). Notwithstanding the IP67 degree of protection, and while adopting the recommendations as regards supporting and positioning the TMSC100, the fact remains that dust, liquids and chips can constitute a hazard to operation of the device and therefore to the reliability of the measurements. Constant preventive maintenance is the best guarantee of reliability. The more the environment is polluted with dust (e.g. graphite dust), the more machining produces lightweight electrostatic swarf and the more coolant is used, the more frequently preventive maintenance should be carried out. It may therefore be termed ordinary maintenance.

Ball Bearing Assembly and Disassembly

The choice of the WIRES's uses case evolved through two different steps. During the first phase a set of possible uses case were identified:

1. wheels and roller strips substitution;
2. substitution of the needle sliding blocks of the carriage;

3. wheels bearing substitution;
4. nitrogen piston substitution.

After the implementation of AR-based maintenance procedures for the above mentioned use cases, WIRES' technicians were asked for evaluating the impact of AR (both using a tablet and a wearable AR glasses) in the maintenance workflow. Unfortunately, they claimed that selected use cases were 'too easy' and AR did not introduce any improvement/facilitation. In order to tackle this issue, the WIRES' use case has changed by selecting more complex maintenance procedures: ball bearing assembly and disassembly.

Roller bearings are a vital and critical part of almost all machines which have rotating parts. Bearing failure leads to safety issues and to high manufacturing costs, because of machine down times, and interruption of production; that is why several alternatives to reactive maintenance of roller bearings have been investigated. The causes of a shorter life of a rolling bearing can be classified as: lack of lubricant, un-correct mounting of the system, un-correct dimensions of the parts related to the bearing, pre-existing defect in the bearing, lack of maintenance procedures.

2.6 User Requirements

User requirements are essential to drive the design step of each architecture. In this case, they have been collected by asking consortium company technicians to answer a questionnaire. The questionnaire (appendix A) was delivered both on-line and by a paper version. Three user categories have been identified: expert, trainee and customer. Moreover, small, medium and large company technicians, from different sectors of work, have been interviewed. The collected requirements had to be ranked/classified according to the 4-level MSCW (MoSCoW) scheme [190]. This use of MoSCoW was first developed by Dai Clegg of Oracle UK Consulting [191]. The description of the four levels is as follows:

- M – Must - the application must satisfy this requirement to be considered minimally functional.
- S – Should - the application should satisfy this requirement to satisfy the essential needs of the users.

Table 2.1 Most rated MUST and SHOULD requirements.

MUST Requirements
Allows the user to move through various stages of the procedure.
Real-time support.
Hands-free.
Offline usage.
Multiple languages.
Menu with a list of available procedures.
SHOULD Requirements
Use of glasses for augmented reality built into helmet.
Don't stop maintenance to obtain information.
The application recognize the environment machine.
View support while performing the procedure (having hands free).
Video Capabilities (recording different formats).
Physical interaction (touch).
Good Connectivity.
Textual assets.
Visual assets.
Not real-time support is dynamic (virtual).

- C – Could - the application could satisfy this requirement if there are sufficient resources.
- W – Would - the application would satisfy this requirement if it was a perfect world and we could have everything we want.

2.6.1 Results

Overall, 17 answers have been received, almost evenly distributed among experts, trainees and customers; the best part of technicians were with medium and large companies, basically working in aerospace and manufacturing sectors. Table 2.1 graphically summarizes the most selected MUST and SHOULD requirements. It is immediately clear that the AR application has to be as flexible as possible. Some requirements are (or seem to be) opposing; for instance, some people hope for a physical interaction but other technicians wish for hands free solutions. The application had to work off-line but a good connectivity is an important requirement. The application should provide textual help as well as visual hints.

2.7 Conclusions

The analysis and investigation described in this chapter constitute the methodology adopted through out all the proposed research. It could be briefly summarized in four steps: analyzing the state of the art; identifying existing problems; determining suitable use cases; collecting user requirements. The next steps in this process are designing and developing solutions for the identified problems and then performing tests to evaluate them. Table 2.2 lists the challenges that will be addressed in the next chapter. Moreover, the analysis presented in section 2.6 allowed to define the following list of issues that are crucial to address:

1. the proposed solution must allow the final user to move through the various stages of a procedure;
2. the user might have both hands dedicated to perform the maintenance task without interruption and the application allows him/her to view information while performing the procedure without interruption;
3. depending on the scene framed by the device, the application displays specific aids such as information about the operations to be performed and the objects involved in the procedure;
4. video see-through and optical see-through should be both available as options;
5. the proposed solution should be environment-independent: tracking should be markerless and independent of the machine surroundings and/or light conditions.

In the next chapter, the first step will be repeated to identify convenient hardware and software solutions that would fit best to deploy the architecture. As prototype solutions are defined and proposed to the testers, steps two and four will be repeated as well, since new problems may arise or new requirements that may be pointed out by the users.

Table 2.2 Challenges

Challenges
Pose Tracking
Reconfigurability
Interface Design
Robustness of the Interface
Training Systems
Technology Acceptance

Chapter 3

Proposed Solutions

Part of the work described in this chapter has also been previously published in [3] and [4], [60], [61] and [62]

The chapter is organized as follows: first, a brief introduction to the hardware and software technologies for Augmented Reality adopted for the proposed work is presented. The following section will illustrate how the results of the analysis proposed in the previous chapter guided the design and development of a workflow for implementing AR-based maintenance procedure. Section 3.4 describes the design and development of the Augmented Maintenance Framework (AMF), a framework that enables the users to adopt the proposed workflow. In sections 3.5 and 3.6, interfaces for AR systems are investigated and some novel solutions are presented.

3.1 Hardware Solutions

The hardware configuration for using HMD devices in AR applications has been presented by Azuma in 1997 [10]. Over the years, researchers came up with other configurations, due to the need of hardware solutions that could suite their applications best. Nowadays, common alternative to the HMD configuration are: the projection-based configuration, the monitor-based configuration and the mobile device-based configuration. This last configuration has become more popular in

Fig. 3.1 Vuzix START 1200XLD

recent years, since it allows the users to move freely in the physical space, without the necessity of wearing cumbersome hardware or staring at stationary screens.

In order to satisfy the flexibility issue raised from the user requirements analysis, two different paradigms have been supported: mobile (i.e., Android tablets and smartphones) and wearable (i.e., AR glasses). It is important to note that wearable does not mean mobile, as AR glasses often require connection to PCs. For the "wearable" category, two different products have been considered: Vuzix Star 1200XLD and Epson Moverio BT-200. The first one has to be connected to a Windows machine (basically a notebook), whereas the Epson glasses are connected to a pocket Android device and therefore it is a real mobile and wearable solution. The following paragraphs briefly list technical characteristics of the two types of AR glasses.

Vuzix Star 1200XLD is one of the most popular first-person AR display solutions available today, compatible with virtually all of the most popular augmented reality development software. STAR 1200XLD Augmented Reality System (see Figure 3.1) uses Vuzix patented quantum optic see-through display technology. It is possible to see the real world directly through its transparent widescreen video displays while computer content is overlaid in full color 2D or 3D. This model currently ships with a 1080p Full HD camera. This particular camera is UVC (Universal Video Class) compliant, meaning it is plug'n play with no additional drivers required for most computer operating systems to function as a standard USB webcam. The STAR 1200XLD comes with a broad assortment of software and accessories. Removable head-tracker with compass, VGA computer interface, rechargeable battery powered interface for composite and component audio/video devices, and the maxReality AR authoring software are included.

Specifications:

Fig. 3.2 Epson Moverio BT-200

- Screen definition: Twin high-resolution WVGA (852 x 480 pixel) LCD displays
- Supports display resolutions of: 640 x 480 (VGA) 852 x 480 (WVGA) 800 x 600 (SVGA) 1024 x 768 (XGA) 1280 x 720 (720p)
- Perceived size: 75-inch virtual screen as viewed from ten feet (3 m)
- Display: 16:9 and 4:3 aspect ratio content
- Field of view: 35 degree diagonal field of view
- Refresh rate: 60 Hz
- color depth: 24-bit
- USB camera: yes
- Video capture: 1Hz to 60Hz
- N° of displays: 2
- N° of cameras: 1
- Wrap Tracker: 3-DOF (degree of freedom) head tracker: yaw, pitch, and roll

Epson's Moverio BT-200 (see Figure 3.2) is an Android-based, see-through wearable display. Featuring a head set with micro-projection technology and a compact Android-powered controller, Moverio allows the user to experience side-by-side 3D content on a “floating” see-through display projected into his environment. As a world leader in projection technology, Epson’s reputation ensures bright, larger-than-life images. The full color displays provide the sensation of watching a 80” diagonal screen at a distance of 5 meters, with a refresh rate of 60 Hz. The product is delivered with head set, controller, 8GB microSDHC card, detachable earphones, lens shade and eye glass holder.

Specifications:

- Screen definition: 960 x 540 pixel
- Perceived size: 80-inch at 5 m
- Field of view: 23°
- Refresh rate: 60 Hz
- Color depth: 24-bit
- Transparency: 70% max
- Dimensions: 178 x 205 x 47 mm
- Weight: approximately 88g for the headset and 124g for the controller
- OS Version: Android 4.0.4 (update via network)
- Wrap tracker: in both headset and controller
- Connectivity: IEEE 802.11b/g/n, Bluetooth 3.0, USB2.0
- GPS in the controller

3.2 Software Solutions

Since the interest for AR applications has raised in the last years, both universities and private companies worked on developing and proposing different AR solutions and software technologies. Among all the available software frameworks, sections 3.2.1 and 3.2.2 describe those who better match the user requirements collected in the previous chapter. Section 3.2.3 describes the comparison performed to select the software adopted for this research and how the user requirements guided the choice.

3.2.1 Commercial Solutions

Metaio

Founded in February 2003, Metaio has been one of the world-wide leaders in Augmented Reality technology, developing software products for visual interactive solutions between the real and the virtual world. The Metaio SDK allows the user to integrate digital content seamlessly into the user's camera view looking upon the real world. Metaio has developed AR custom solutions for different industries such as: IKEA, Macy's, Toyota, Universal Pictures, Volkswagen, Adidas, BMW, Audi, Bauer Media, USA Today, Bosch, Mitsubishi Electric and LEGO.

Vuforia

Vuforia is an Augmented Reality SDK for mobile devices that enables the creation of applications that can recognize and track both planar images and simple 3D objects, such as boxes, in real-time. Additional features of the SDK include localized occlusion detection, runtime image target selection, and the ability to create and reconfigure target sets programmatically at runtime. The SDK supports both native development for iOS and Android while also enabling the development of AR applications in Unity that are easily portable to both platforms. Application Programming Interfaces (API) are available for C++, Java, Objective-C++ and the .Net languages.

3.2.2 Institutional Solutions

Graz University of Technology

The Graz University of Technology is one of the largest public university in Austria and one of his main buildings is dedicated to the Institute for Computer Graphics and Vision, founded in 1992. Its research is focused on Virtual Reality, Augmented Reality, Medical Computer Vision, Object Recognition, Object Reconstruction, Robotics. The institute has many research groups focusing in different fields of

Augmented Reality, such as AR Visualization, AR Tracking and Mobile AR. The last one was the first to introduce Augmented Reality on PDAs in 2003 and the first to demonstrate real-time Natural Feature Tracking on mobile phones in 2007.

One of the Institute's major research project is Studierstube, started in 1997. The Studierstube project has the purpose of delivering a framework for the development of mobile, collaborative and ubiquitous AR applications. The motivation behind the project was the belief that augmented reality could have a better chance of becoming a viable user interface for applications requiring manipulation of complex three-dimensional information as a daily routine.

Georgia Institute of Technology

The Georgia Institute of Technology is a public research university in Atlanta, Georgia, in the United States. Its Graphics, Visualization & Usability Center (or GVU) researches in areas of study such as human-computer interaction, information visualization, ubiquitous computing, virtual reality and augmented reality. The most important project related to AR is the DART project. DART (The Designer's AR Toolkit) is aimed at enabling designers to work directly and effectively with AR. The design of DART, resulting from a tight collaborations with designers, addresses a collection of problems that, together, make AR a difficult medium to work with.

VTT Technical Research Centre of Finland

VTT Technical Research Center of Finland is one of the largest multidisciplinary non-profit research organization in Northern Europe.The VTT Augmented Reality team designed the ALVAR framework to offer high-level tools and methods for creating augmented reality applications with just a few lines of code. The library, that is released under the terms of the GNU Lesser General Public License, is currently available on Windows and Linux operating systems and only depends on one third party library (OpenCV). Furthermore, the ALVAR Mobile SDK for iOS, Android, Symbian, Maemo, Flash and Silverlight platforms, as well as the ALVAR Render engine, are available for internal use and selected partners.

Seac02

Seac02 is an Italian company with his headquarter located in Turin which realize the LinceoVR framework. This framework allows the immediate contextualization of the product created in the environment, thanks to instantaneous camera matching. Texture tracking is also available in camera matching mode, allowing users to choose their own pictures as the targets to be tracked. LinceoVR's AR module supports any kind of webcam or professional camera available in the market with the easiness of 'plug and play': the 3D model is rendered with a solid color background while the hidden video is used for the tracking and to move the 3D object. Thanks to this feature, LinceoVR can be easily used with See-Through Glasses for Augmented Reality applications, such as the Vuzix iWear VR920 and CamAR.

DroidAR

DroidAR is another framework for Augmented Reality, specifically designed for Android. Since the creation of its first prototype in 2010, DroidAR was continuously improved and new features were added through the years. The proposed development tool (SDK) enables android programmers to integrate AR in their own applications.The SDK is provided without charge with an Open Source License for non-commercial projects. The feedbacks from the developers as well as its application in seminars and practical project seminars greatly aided its improvement, turning this framework into one of the most used open source augmented reality SDKs for the Android platform.

3.2.3 Software Comparison

In order to choose the software framework to be adopted for this research, a comparison among all the most important existing ones has been performed. Figure 3.3 offer a platform-based classification of existing augmented reality SDK whereas the table available in appendix B shows a comparison of all the available AR software solutions (at the time of this analysis). The choice of the software framework was based on the following criteria, taking into account the analysis and the user requirements described in the previous chapter:

- compatible operative systems;

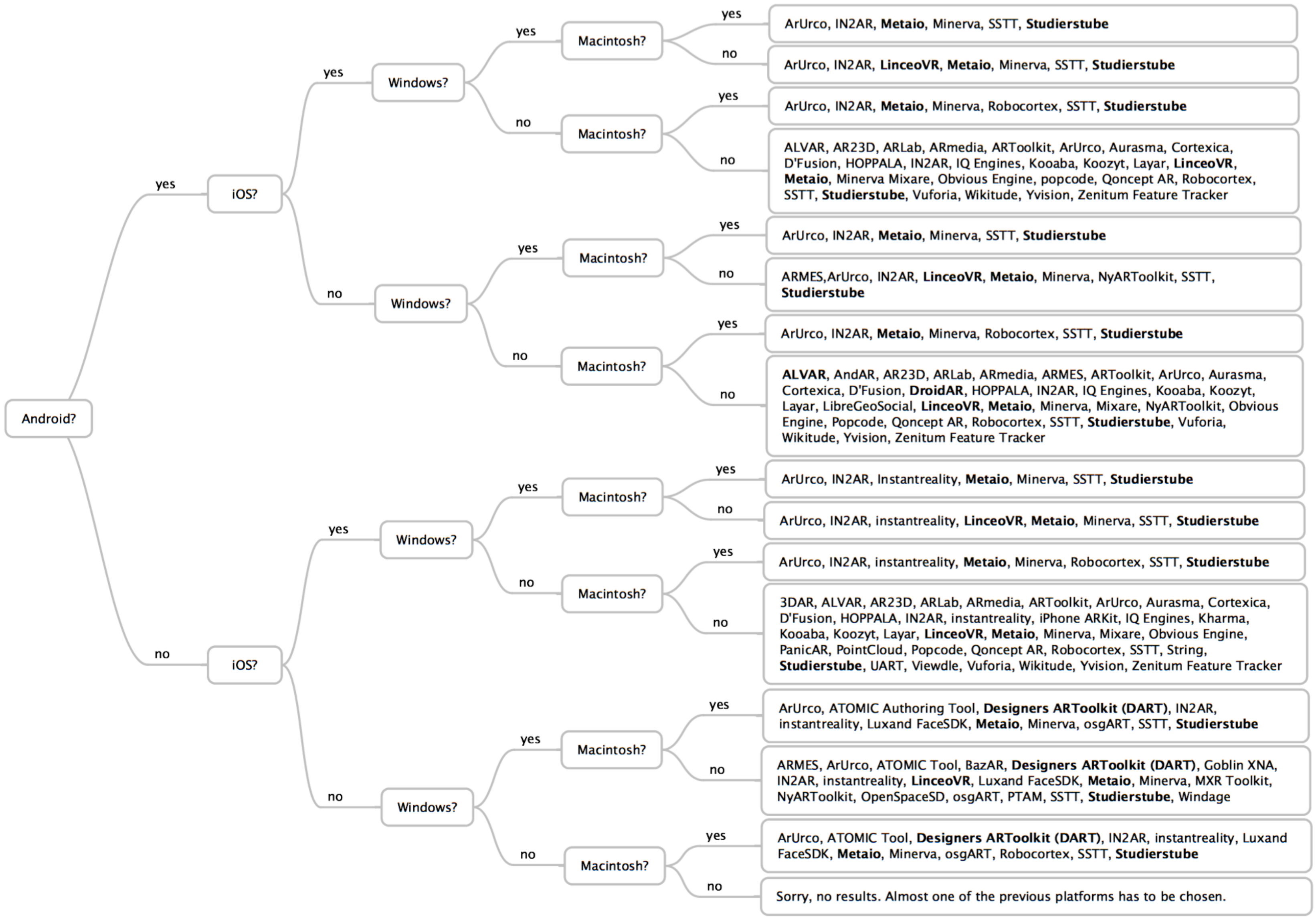

Fig. 3.3 AR SDK Platform-based classification.

- available types of tracking systems, such as marker tracking, image tracking, 3D tracking, etc.;
- API availability;
- distribution license.

When this analysis was performed (2013), Metaio was the only framework that provided CAD tracking, that is essential to provide an environment-independent solution. Moreover, all the necessary software packages and libraries were available through a free license and all the operative systems compatible with the chosen hardware were supported. For these reasons Metaio SDK was chosen as the reference software framework for this research. At present time, Metaio has been acquired by Apple and the Metaio SDK is no more available to the public.

3.3 A Workflow for Implementing Maintenance Procedures

This section aims to describe all the steps to be performed in order to design and develop an augmented reality (AR) procedure. This document gets in input the analysis of technologies, the study of the state of the art and user requirements. In particular, the software solution provided by Metaio stands out from the analysis of AR technologies and it has been selected for developing the AR applications for the project. Although the procedures described in this document refer to Metaio, several steps are absolutely general and should be considered also if another technology (e.g., Vuforia) were considered.

Developing an AR application for maintenance, repair and assembly needs to consider several issues and problems. First of all, an efficient tracking system has to be identified. It can be based on the object (or objects) to be tracked or it can be based on artificial features (e.g., markers), which can be added to the object itself. Moreover, the focus can be either on just one tracking system or on a hybrid approach, depending on steps of the procedures to be performed. For example, it can be necessary to recognize an object with a complex geometry (by using CAD or 3D Map tracking) and then a flat surface with several textures (by using 2D image tracking).

Depending on the tracking system, a set of configuration files needs to be created. Configuration files can be several 2D images, such as a photo or computer-generated renderings, or 3D models of part of the object to be tracked.

The next step involves the description of the procedure to be used for supporting the end-user during the maintenance/assembly task. Computer-generated (sometimes called virtual) hints can be:

- text labels;
- images;
- 3D static models;
- 2D and/or 3D animations;
- videos;
- audio messages.

During the last step, all the code to perform the above mentioned contents (and possibly extra features depending on user requirements) has to be implemented for the selected AR technology. The diagram showed in Figure 3.4 provides a high-level visualization of the workflow for developing an AR application.

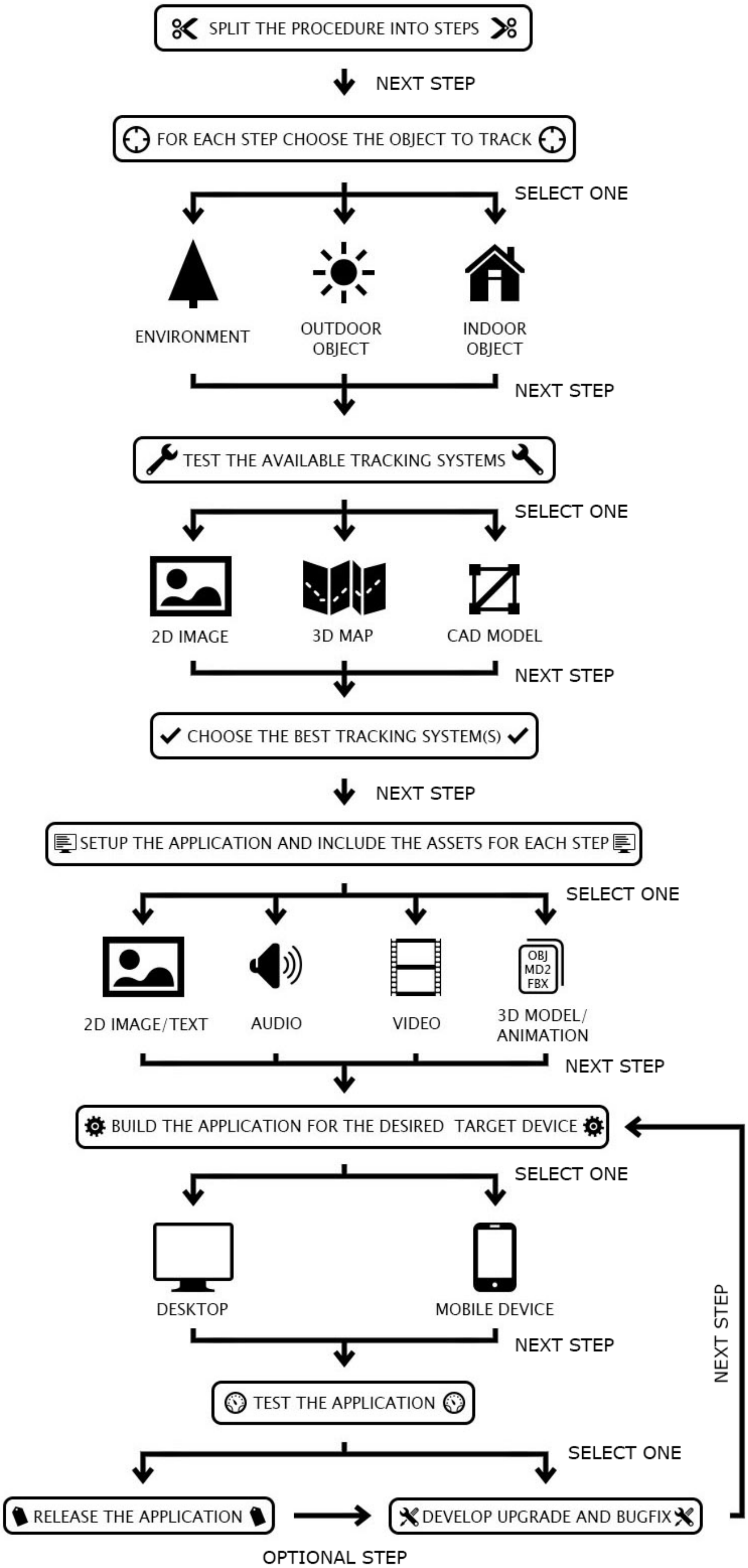

Fig. 3.4 The workflow diagram.

3.3.1 Resources

This section provides a list of software and libraries useful to develop the application, create the assets and the tracking configurations.

Metaio SDK

Metaio SDK is an Augmented Reality Software Development Kit that allows users to develop AR applications for both mobile devices and computers; moreover, it provides a wide choice of tracking systems. The SDK is currently supported on Android, iOS and Windows with an additional plug-in for development in Unity3D for Android, iOS, Windows and OS X platforms.

Metaio Creator

Metaio Creator is an Augmented Reality software that allows users to create a complete AR scenario without special programming skills. It is well suited for quick prototyping and for testing out different tracking patterns for a given object.

Metaio Edge Config Tool

This tool, included with the Metaio Creator package, allows users to create a line-like representation of the external wireframe of a 3D model (formerly 'line model') for a given view. This tool is useful for implementing a CAD-based tracking.

Rhinoceros

Rhinoceros [192] (or simply Rhino) is a commercial NURBS based 3D modeling software, developed by Robert McNeel & Associates. The software is commonly used for industrial design, architecture, marine design, jewelry design, automotive design, rapid prototyping, reverse engineering, product design, as well as the multimedia and graphic design industries.

Autodesk 3D Studio Max

Autodesk 3Ds Max [193], formerly 3D Studio Max, is a 3D computer graphics program for making 3D animations, models, and images developed and produced by Autodesk Media and Entertainment. It has modeling capabilities, a flexible plug-in architecture and can be used on the Microsoft Windows platform. In addition to its modeling and animation tools, 3D Studio Max also features shaders (such as ambient occlusion and subsurface scattering), dynamic simulation, particle systems, radiosity, normal map creation and rendering, global illumination, a customizable user interface, and its own scripting language.

Autodesk Inventor

Autodesk Inventor [194] is a 3D mechanical CAD design software for creating 3D digital prototypes used in the design, visualization and simulation of products.

Blender

Blender [195] is a free and open-source 3D computer graphics software product used for creating animated films, visual effects, interactive 3D applications or video games. Blender's features include 3D modeling, UV unwrapping, texturing, rigging and skinning, fluid and smoke simulation, particle simulation, soft body simulation, animating, match moving, camera tracking, rendering, video editing and compositing. It also features a built-in game engine.

Metaio Toolbox

Metaio Toolbox is an iOS/Android application, empowering the user to create or edit 3D tracking maps of all textured objects in his/her surrounding. The user can further use the created maps in Metaio SDK, through Junaio or load it to the Metaio Creator to build an AR-application. Furthermore, the Toolbox also has camera calibration function that allows the user to determine camera parameters of his/her specific device.

3.3.2 Tracking Techniques

2D image tracking

This tracking system is based on providing an image to match with the current frame from the camera. The image could either be a photo of the object or render of a 3D model of it. Depending on the provided image, environment would be part of the recognition process, but could be excluded for better reusability of the application. The application recognizes the object when the camera frame of the object matches the provided image, with the same point of view. The recognition will go on till the system could ensure that the tracked object is inside the area of the camera, even stretching the camera view (to some extent). Metaio SDK currently offers two kinds of marker less image tracking: one Fast and one Robust method. The Fast method works well on a wide variety of targets. It runs quite fluently on most relatively recent smart phones and it is very stable on moderately textured targets. The Robust method should be tried on targets where the Fast method did not provide satisfactory results. The Robust method fits well to highly textured targets. The tracking results obtained with the Robust method improve over time: the more and the longer the user moves the mobile device in front of the tracked target the better the tracking gets. In Markerless Configuration, markerless image tracking is explained in more detail. For markerless image tracking, there are currently only two main parameters in addition to the reference image file that is used for tracking. In particular, the parameters “widthMM” and “heightMM” specify the size of the printed reference image that will be used for tracking. These two parameters are important to correctly scale the virtual assets to the real world.

3D map tracking

The 3D maps markerless tracking allows the user to choose any real world object as a tracking reference. To use a 3D map tracking configuration, the user should create a configuration with the Toolbox app and load the output configuration. The output of the process is a .3Dmap file, which is intended to be used with the Creator or the SDK. This technique is based on creating a map of point with the Toolbox app. This map will be used in the developed application to match the current viewed frame to check for positive recognition. The points created by the mapping are based on geometry and texture, and could come from both the object and the environment.

During the making of the 3D map, all of the visible features in the proximity of the object are detected from the smartphone camera. For this reason, it's better to manage the entire operation on a neutral background, avoiding any undesired interference with the surrounding environment. The map could be edited to improve the robustness, removing useless point and trying to focus on the object itself. As the texture information is used for the tracking, light condition and texture of the objects should be considered and could provide limitation tracking the object.

CAD tracking

3D Markerless Tracking based on CAD data allows a precise 3D pose localization based on a given 3D model of (a part of) the environment, for example a small object that should be tracked or a building to which the camera pose should be determined. It is used much like extensible tracking, while using a 3D model for an edge based pose initialization to enable an accurately scaled and localized augmentation. It detects the camera pose only once and then switches to Markerless 3D tracking.

The edge based initialization process uses a separately controllable Dummy Tracker or GPS, compass and gravitational sensor information as a prior pose approximation and searches the correct camera pose in a specifiable range around it. It requires a surface model and a line model of the environment or object to track, either getting it through the Metaio Edge Config Tool for a single view or creating it with a modeling software or automatic algorithm from the original 3D model.

3.3.3 Creating Assets

Starting from the object that the user wants to track, it is necessary to create the proper assets to instruct the user about the maintenance procedure. The Metaio SDK offers content creation in three major areas: images, movies and 3D animations. With the help of Metaio SDK, developers can easily manipulate image geometries according to the requirement of the application. It is also possible to apply movie textures to a tracking target. As simple as setting up image geometries, developers can create movie geometries and explore different configurations for movie playing with a few simple lines of coding. The Metaio SDK also offers developers tools to convert video files so that they can be played smoothly in an AR setting. To achieve a full-blown AR experience, the Metaio SDK can be fully functional in conjunction with 3D modeling and animation software. 3D content can be either created using 3D animation software or used directly off-the-shelf. As long as it is in a format that Metaio SDK supports, the 3D content can be easily exported to AR applications.

Images

Using image content for mobile applications is the simplest form of Augmented Reality. A major rationalization for overlaying an image on or around a specific target is to offer additional information or change the appearance of the original target. For example, in a GPS-based application using GPS sensors to provide information on geo location to the tracker, the developer can simply enclose more detailed description of the geo location with, for example, a billboard. This is very useful in a navigation system that helps the end user to identify the places that are of great interest. In a virtual clothing store application, on the other hand, the primary objective is to let the end user try on different clothes in an AR environment. For these types of applications, the developers can simply employ the built-in functions within the Metaio SDK to overlay new clothing images onto the tracking target–the customer. Additionally, the developer can also set the scale and translation parameters to allow manipulation of the clothing images by the end user.

Currently, the Metaio SDK supports three image formats: .jpg, .png and .bmp. The resolution of the image texture can be flexible since the developer can alter the scale of the image using the SDK. Apart from the manipulation of image scale, the translation parameter can also be modified to adjust the location of the image texture against the tracking target.

Movies

Movie texture is more advanced 2D content to be employed within Metaio SDK. With the built-in functions, the user can specify the path of a movie and overlay it on a 3D plane over the tracking target. It should be noted that by movie texture, it means that the movie is a full video file instead of a streaming video. The movie should be playable offline on mobile devices. Metaio SDK imposes strict specifications on the codec and format of the movie texture to ensure smooth playback on mobile devices. It is the developer's responsibility to ensure that the movie texture fits the requirements enforced by Metaio SDK. Here are the specifications for video files:

- Video Compression: MPEG4 codec inside 3G2 container.
- Video Resolution: recommended 176x144px @ 20fps (e.g. 288kbps) - more powerful devices (such as dual core devices) could handle higher resolutions
- Audio Compression: AAC LC
- Audio Resolution: 22050kHz Stereo (e.g. 48kbps)
- Aspect Ratio: Supplied geometry must have same aspect as the encoded movie

Videos can be converted by Super (http://www.erightsoft.com/SUPER.html) to the format supported by Metaio SDK. Alternatively, the user can also adopt the following online video conversion tool to do the job: http://video.online-convert.com/convert-to-3g2.

3D & Animations

Poly(gon)count is defined as the total number of polygons in a 3-D model. Usually, this numbers unit is counted in triangles. A high-resolution model with a lot of polygons is called "high-polymodel" in comparison to a "low-poly-model" with a small polycount. Sometimes these categories are not clearly distinguished. They are dependent on the device and the usage of the models. In case of real-time applications mainly low-poly-models are being used. These models render faster and increase the performance of an application. Currently Metaio SDK supports OBJ for static objects, MD2 and FBX for animated objects.

The OBJ file format is a model format for static meshes. It supports multiple meshes with different textures and materials. Meshes can also have assigned different materials to their polygons. Thereby the OBJ format provides more flexibility with respect to textures than the MD2 format which can only use one texture. Thus, it is recommended to use the OBJ format for high quality static objects and rely on the MD2 format for animated models.

The MD2 file format is an ancient but efficient 3D file format which nicely supports vertex animations. MD2 models cannot have different colors but only one texture. Therefore, textures have to be baked in case for instance "vertex colors" shall be shown. Moreover, the filename of the texture must be the same as the model's (e.g. modelFile.md2 and modelFile.png). One MD2 file can contain a set of different animations. The animations are calculated on per-Vertex base and are very efficient for mobile devices with weak processors. Although the MD2 file format is a vertex-animation format, the file itself is very small in size compared to other vertex-animation supporting formats (like VRML for example). A model with around 500 vertices, 1000 polygons and a total frame number of 200 with all animations will require about 480 KB of storage on the mobile device. When this exemplary model is loaded into the renderer it will consume additionally 750 KB of system memory. Usually MD2 models do not need to have more than several thousand vertices. There are several models ready for download which can be used with the Metaio Mobile SDK. The following free Viewers can be used to review the md2-models and animations:

- Fragmotion
- Quake 2 Model Editor v. 0.90;
- Misfit Model 3D - a free all around tool it is worth looking into. Very nice for adjusting animations of md2s.

The FBX file format started as a motion capture format but has since become the all-round data exchange format for most of Autodesk's products. It is also supported by a wide range of tools in the 3D modeling domain. It might contain almost everything from 3D model data to animation or even movie data. However, to fit the capabilities of today's mobile devices, the Metaio SDK currently only supports a subset of the numerous features. Namely the Metaio SDK supports 3D model data (multiple meshes need to be grouped), one diffuse texture per material, diffuse vertex colors and multiple named skeletal or keyframe animations. This is to overcome the limitations of OBJ (no animations at all), MD2 (vertex animations only, limited vertex count, bad accuracy or wobbling) and a wide range of tools with appropriate export filters. In order to make FBX suitable for mobile devices, an additional conversion step is needed to strip all unwanted items from the file and turn it into a fast runtime format. An easy-to-use tool named FBXMeshConverter to perform this step is provided as a part of the SDK and a separate download. Metaio SDK supports the most recent version (2013) of FBX files but is backwards compatible. To have multiple named animations in a single FBX file we found Autodesk MotionBuilder and Blender the most useful tools with integrated multi-take support.

3.3.4 An Example of AR Procedure Creation

The following case study shows the development of an AR application for maintenance procedure on an ink-jet printer. The application has been developed to test the CAD and 2D tracking system by the Metaio framework. 3D Map tracking was avoided because it intrinsically relies on texture information and light condition, whereas the purpose of this research is to investigate solutions that are independent of the environmental conditions.

Workflow Diagram

The first step is to define a diagram that describes all the steps to be performed to develop the application, starting from the specifications. The main purpose of this application is to provide instructions for changing the two toners (black and color) of an ink-jet printer. A 3D CAD model of the printer is necessary to implement the CAD tracking by the Metaio framework. Then, the components of the printer used as assets are animated. The next step develops the application with Metaio Creator. Finally, the application is tested to verify the reliability of the different tracking systems implemented.

Modelling the Object

A 3D mesh of the printer is modelled in Blender as similar as possible to the real printer. The model will be used for:

- real-time tracking recognition with the CAD system;
- rendering some poses of the model to test the reliability of the tracking system;
- creating the animation that will be used as assets in the application.

For providing a better visualization of the assets in the final application, the texture baking procedure is used (Figure 3.5). This procedure allows the user to: define a texture for the model, setup a lighting/shading system in Blender and perform an UV mapping of the texture that includes the light parameter. This feature is not used by the CAD tracking system, because it relies only on the wireframe of the model.

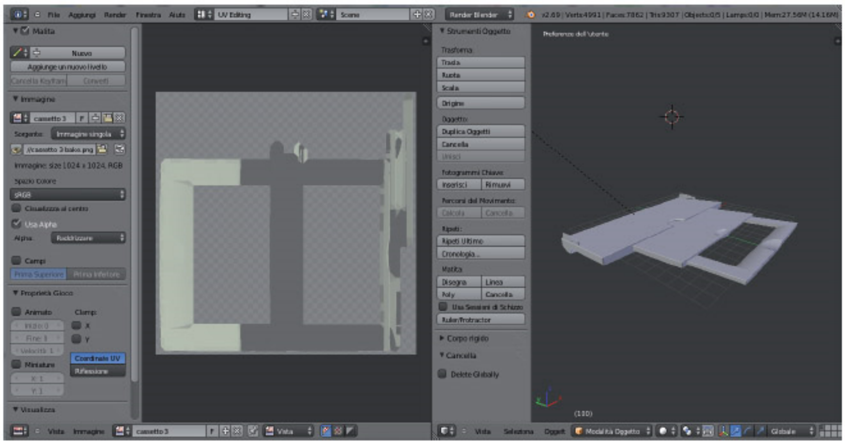

Fig. 3.5 The texture baking procedure: a texture is mapped by the UV mapping system to add texture, lightning and shading to render the object.

When the model is ready, it is necessary to setup the poses that identify the steps of the procedure. In this example, four different poses of the printer define the procedure (Figure 3.6). Each pose, is then exported as an .obj file, the format required by Metaio Creator.

Rendering the Poses

The four poses showed in Figure 3.6 are rendered to perform two tasks:

- testing the CAD tracking system,
- developing a 2D Image tracking system.

The final application has to be able to perform the maintenance procedure using the rendered poses as references, thus making the tracking system independent of the lighting/shadowing of the object, textures color or materials applied to the object. Another possibility is to use the rendering of the four poses to test the 2D image recognition system offered by the Metaio framework.

Creating the assets

The last step to obtain all the resources for developing the application is to create the assets. In this case study, the assets consist of a set of animations. Each

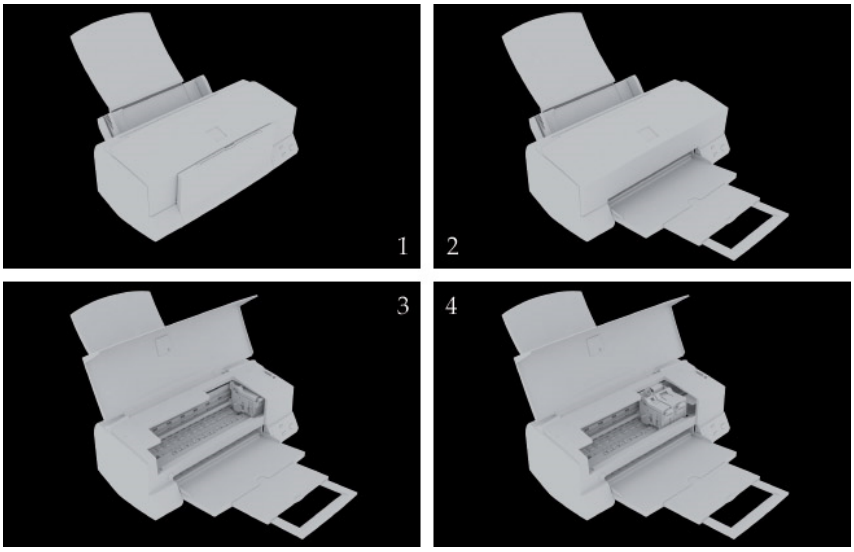

Fig. 3.6 The four poses defining the four different steps of the maintenance procedure.

animation represents one of the task that the final user should perform for the specific maintenance procedure. The following animation are implemented in Blender:

- top panel open/close;
- front panel open/close (Figure 3.7);
- toner case translation left/right;
- toner case open/close;
- extraction and insertion of the toner;

When all the animations are ready, it is necessary to convert the resulting FBX files with the FBX Mesh Converter tool provided by Metaio. The output will be an MFBX package compatible with Metaio Creator.

Creating the application with Metaio Creator

After starting Metaio Creator and creating a new project, for each pose defined for the procedure, the following steps have to be performed:

1. creating a new scene;

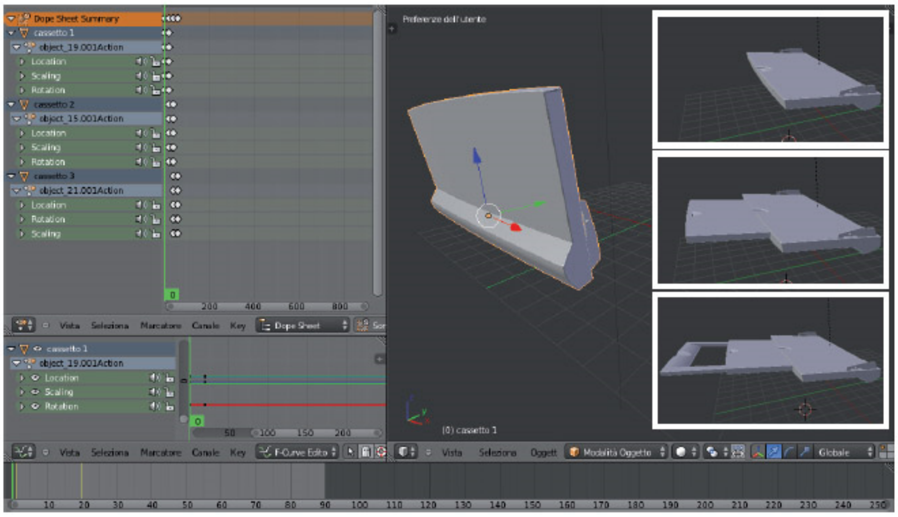

Fig. 3.7 The front panel animation implemented in Blender.

2. importing and positioning in the scene the .obj file that represents the pose to be tracked (Figure 3.8);
3. creating a line object model through the edge tool utility (Figure 3.9);
4. tuning parameters defining the tracking system;
5. importing the assets and align them with respect to the object;
6. tuning parameters defining the assets behavior;
7. testing the animation with the preview.

When all the scenes are ready, it is possible to export the project as an Android application. Opening the project with an Android IDE, the application is compiled as an .apk package and uploaded to the smartphone/tablet device for testing. To develop another version of the application that relies on the 2D Image tracking system, the previous steps are performed with the following differences: in step 2 renderings are imported as .jpeg files, instead of the .obj; step 3 is not performed as it is specific for the CAD tracking system.

Testing the Applications

The final step is to test the application on the mobile device. To have an extended set of test cases, the following tests are performed (Figure 3.10):

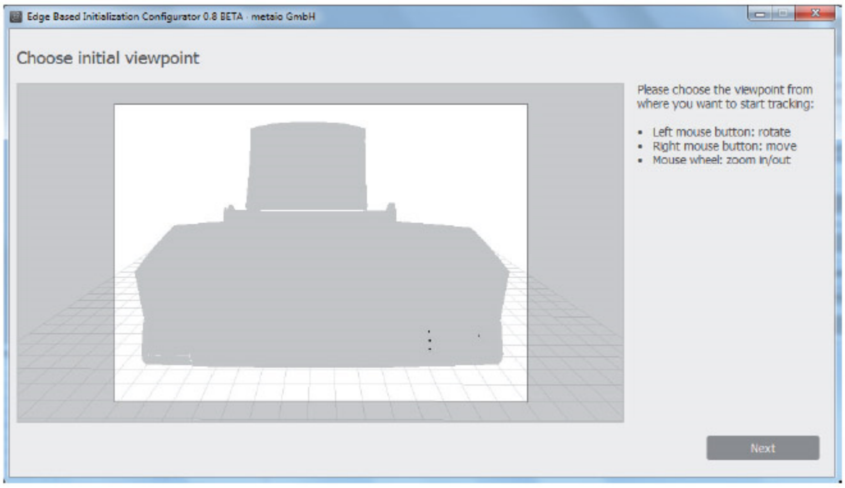

Fig. 3.8 An example of tracking object model.

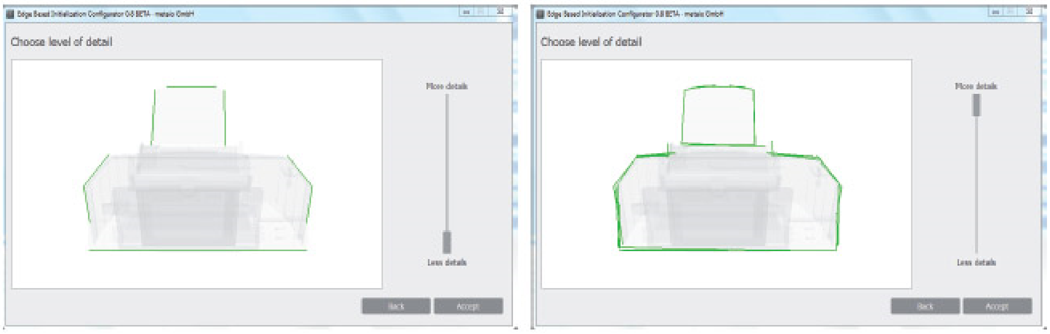

Fig. 3.9 An example of line model object (wireframe) representation.

- test on the real object;
- test on photos of the real object;
- test on the rendered poses;
- test on the 3D model.

The data gathered by the tests are elaborated to improve the tracking system, iteratively working on steps 3, 4 and 6 of the Metaio Creator procedure.

Fig. 3.10 Different configurations to evaluate the tracking solutions.

3.4 System Architecture

It is important to outline how the main constraint of AR applications is basically due to the “rigidity” of contents: it is very difficult for AR users to effectively change the logic of AR solutions. Therefore, the goal of “flexibility” led the design of the proposed architecture. In order to support technicians when they perform at the client sites, a client-server architecture has been chosen: the server side represents the headquarter where an expert can dynamically support remote clients (maintainers). The scene framed by the camera of the client’s device is transmitted to the server, thus allowing the expert to have the same point of view of the maintainer. Moreover, a full-duplex audio communication channel is established, thus providing a sort of tele-assistance. The server application allows the expert to dynamically reconfigure AR procedures by adding/deleting steps, by adding/deleting assets and by changing tracking configurations. At the client side, both Android mobile devices and special purpose AR-glasses are supported; in this way, each company can find the best trade-off between costs and performance of the proposed framework.

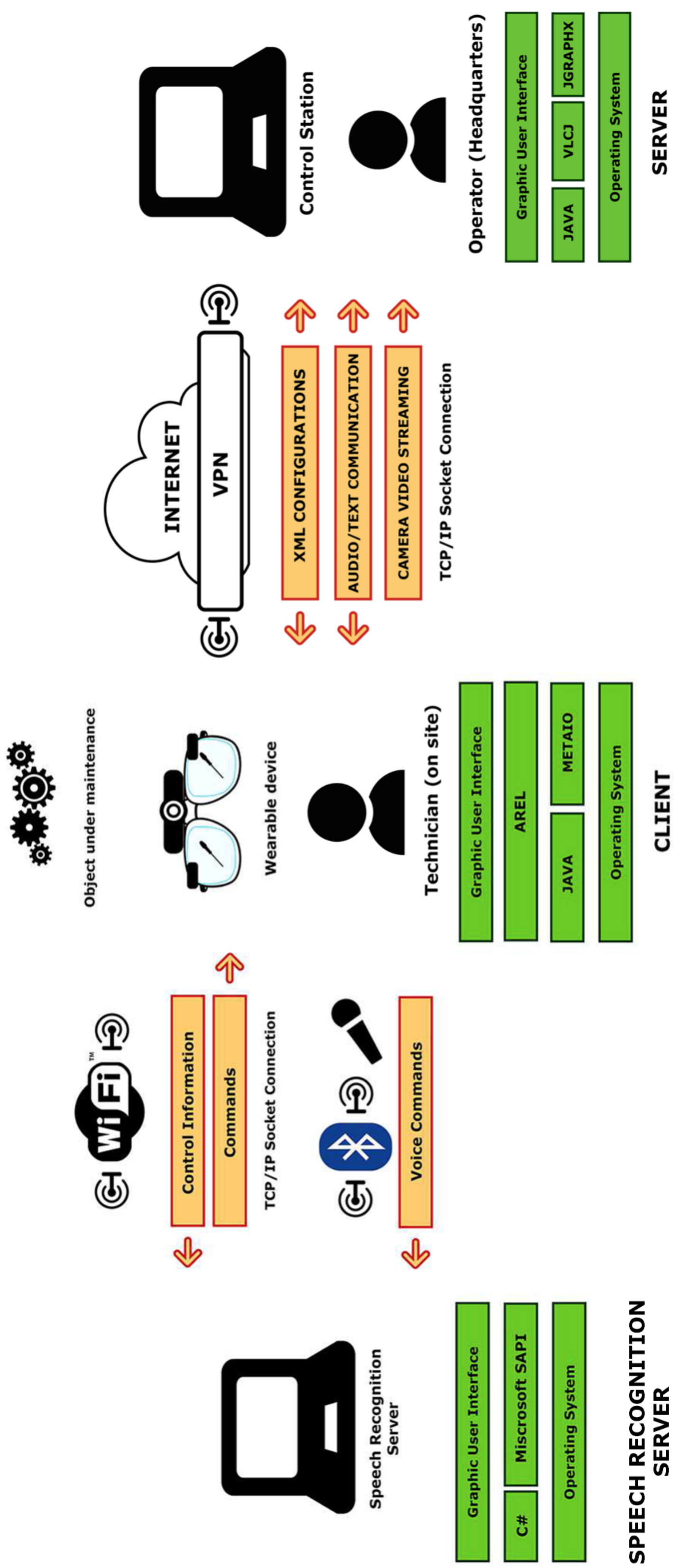

Fig. 3.11 The proposed framework.

CHALLENGES	PROPOSED SOLUTION
POSE TRACKING	CAD Tracking.
RECONFIGURABILITY	State machine representation of the procedure.
CONTENT CREATION	Content creation application.
INTERFACE DESIGN	Framework to develop intuitive interfaces.
INTERFACE ROBUSTNESS	Investigation of vocal one switch interfaces.
TECHNOLOGY ACCEPTANCE	User requirements collected from maintainers. Maintainers have been involved in the test sessions to assess the proposed solutions.

Fig. 3.12 The proposed solutions.

Figure 3.11 provides an overview of the overall architecture. This section will focus on the AMF client-server architecture, whereas section 3.5 and section 3.6 will describe into details the architecture pertaining the interface generation system and the speech recognition system. Figure 3.12 illustrates how the challenges depicted in the previous chapter have been addressed through the architectural choices.

3.4.1 AMF Architecture

The client side consists of an AR application and provides the maintainer with a sequence of steps to be performed in order to accomplish a well-defined task. The application is available in two versions, both developed with the Metaio SDK: an Android application for mobile devices and a Windows application for AR-glasses, which relies on Windows drivers. This first application is intended for a better mobility, compatibility and costs, as it runs on a generic Android mobile device, whereas the second option allows hand-free operations for better performing maintenance procedures. Software layer architectures both of the server and clients are provided in Figure 3.13.

The server side is implemented by a Java application, which runs on both Windows and Unix O.S.: this remote station allows a technician at the headquarter to give support to on site maintainers when they request assistance, through the following functionality:

- sending a procedure to the client;

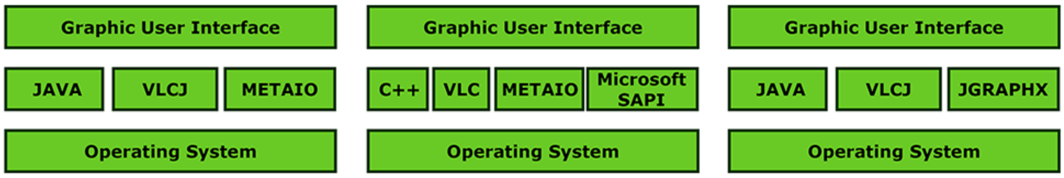

Fig. 3.13 Software layer architectures for the Android client (left), the Windows client (middle) and the server (right).

- modifying and sending back to the client the procedure currently performed by the maintainer;
- communicating with the maintainer through the developed framework;
- sending the camera stream to the server.

The server side application is conceptually composed by 7 modules:

- procedure management;
- reading and writing of a procedure (an XML file is managed);
- assets and tracking configurations loading;
- state machine management;
- user interface (and GUI interaction);
- management of graphics preview;
- connection with client.

The class diagram shown in Figure 3.15 represents the implementation of the server application. Controller is the main class and it is the overall manager. Main attributes of the Controller class are:

- GUI: it is an instance of the UserInterface class, that is the object corresponding to the graphics interface of the application.
- currentProcedura: it is an instance of the Procedura class, which represents the prococedure currently under analysis. The procedure can be already existing or created from scratch.

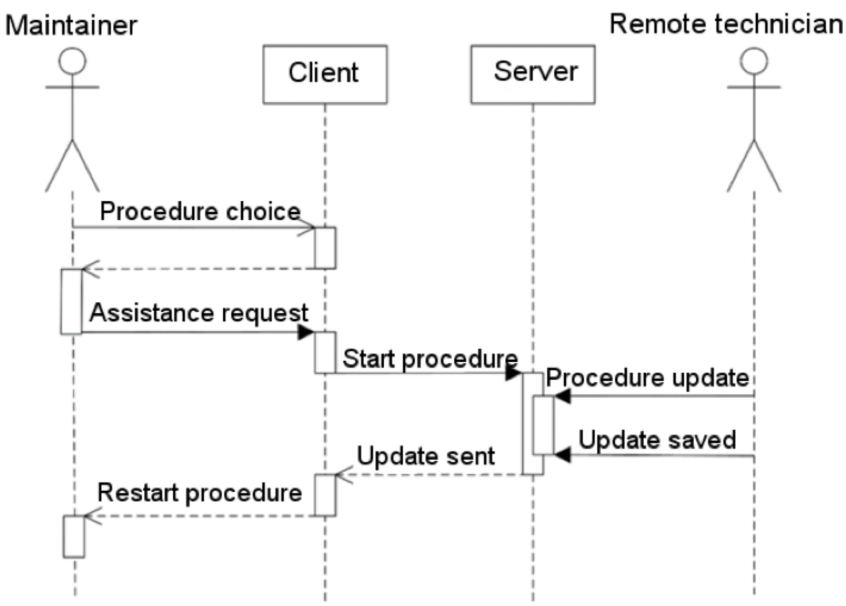

Fig. 3.14 The procedure's update steps.

- graph: it represents a procedure as a sequence of states where arcs are event (that is a tracking configuration has been recognized) and nodes are procedure steps.
- server: it is an instance of the ServerConnection class, which allows the server to establish TCP connection with clients.

The connection with a client is managed by the ServerConnection class. To establish a connection, the server has to open a connection and it has to be waiting for a connection request from a client. Then, interactions between server and client go on as described in Figure 3.14.

3.4.2 Client-Server Interaction

When the maintainer runs the client side application, a list of available procedures is displayed. A remote assistance connection allows the maintainer to request new procedures or update the current one. When a procedure is started, at each step, the real object has to be framed by the device camera: a silhouette representing the tracking configuration has to be aligned to the real object (the silhouette appears as a transparent 3D model on the user interface). When the tracking engine of the appli-

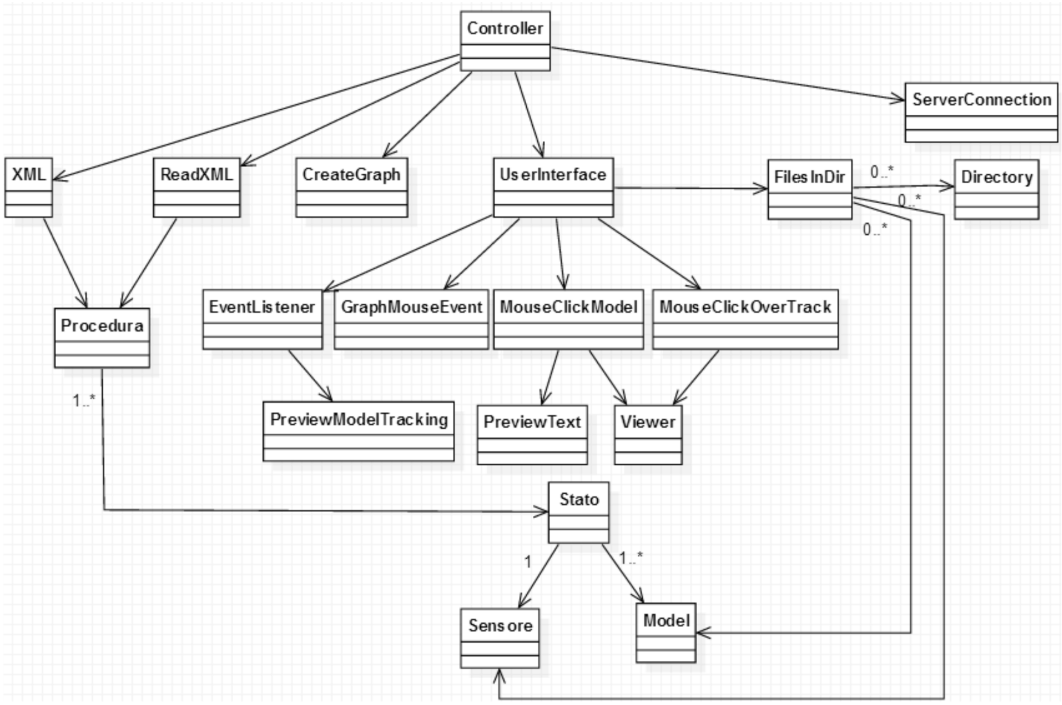

Fig. 3.15 The Client-Server interaction.

cation recognizes the corresponding CAD model (e.g. the tracking configuration), visual and audio assets are provided to the user. When the current step is completed, the maintainer can move on by the 'next step' command. Then, the tracking engine looks for the new tracking configuration correspondence in the scene. The choice to offer two explicit commands to freely move back and forth through procedure steps is for offering maintainers more flexibility; for instance, the maintainer can skip steps when computer-generated aids are not necessary or move back to check (and possibly fix) problems. Moreover, this choice limits the number of false positives in the tracking recognition process, as the maintainer has to confirm to be ready to move on to the next step of the procedure. If the recognition of the next step does not work properly, it could be an operation that the maintainer has forgotten to perform in a previous step. If the maintainer needs to interact with the remote technician, the communication command allows him/her to establish a communication channel. This function connects the AR-application to the technician's remote station (usually over a Virtual Private Network, VPN), opens a full duplex audio channel and sends the video framed by the maintainer's device camera to the remote technician. Data is sent over a TCP/IP socket and the procedure's format is an extension of the XML schema defined by Metaio, which contains all information about the tracking configurations and asset visualization, as they are necessary to describe the machine state diagram. Procedures are coded by XML files and they are already stored on the client device. Each XML file contains all information about the tracking configurations and asset visualization, as they are necessary to describe the machine state diagram. At the beginning, a procedure ID is exchanged between client and server in order to inform the server about the procedure chosen at the client side. When the remote technician receives the state diagram of the procedure, the current step of the procedure is highlighted in the graphic interface, in this way, the remote technician is informed about the step the maintainer is actually performing. Through the audio communication channel the maintainer may request a specific help to the remote technician, thus underlining inconsistency between the procedure and the real case or requesting an explanation about the operation to perform. Eventually, the remote technician may need to fix/update the procedure and send it back to the maintainer. The application at the server side allows the remote technician to change assets, tracking configurations, remove/add nodes and edges in order to provide a new and better set of instructions to the maintainer. When the reconfiguration is completed, the remote technician can send it back to the maintainer, which can move on with the procedure starting from the last performed step. Figure 3.14 shows the event sequence when a procedure update is necessary.

AR-based Training System for Industrial Maintenance

The proposed AMF can be also used to provide AR training: the interaction between the instructor and the student, depicted in Figure 3.16, is similar to the client-server interaction described in the previous section. It can be split in three steps:

1. provide the procedure to the student;
2. execute the procedure and interacting with the instructor;
3. modify the procedure and resubmit it to the student.

This approach has been chosen to maximize the flexibility of the whole system: the instructor can easily produce a procedure for students, interact with them during the practice and update the procedure on the fly, on the basis of students' skills, feedbacks and real-time depending variables.

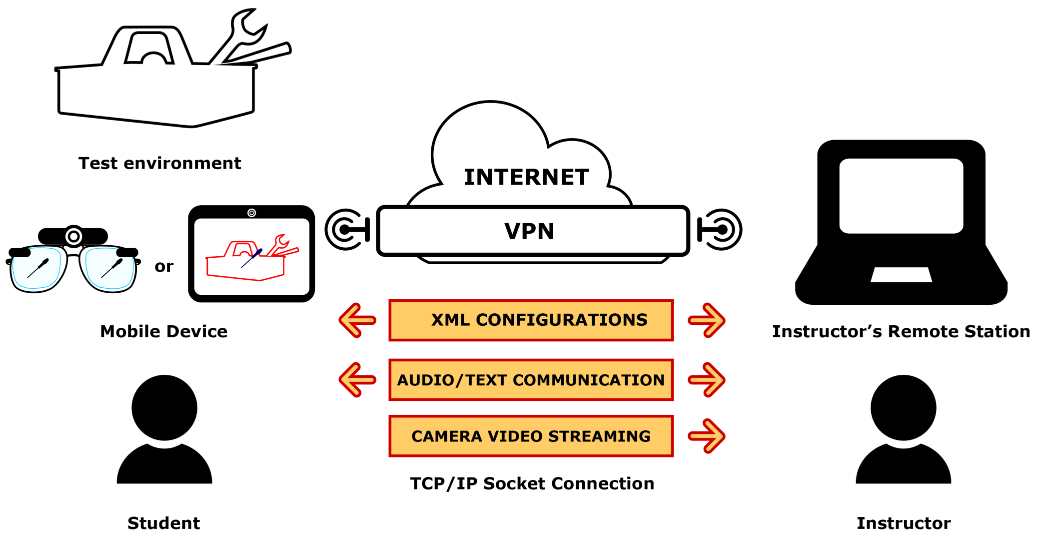

Fig. 3.16 Student-teacher interaction.

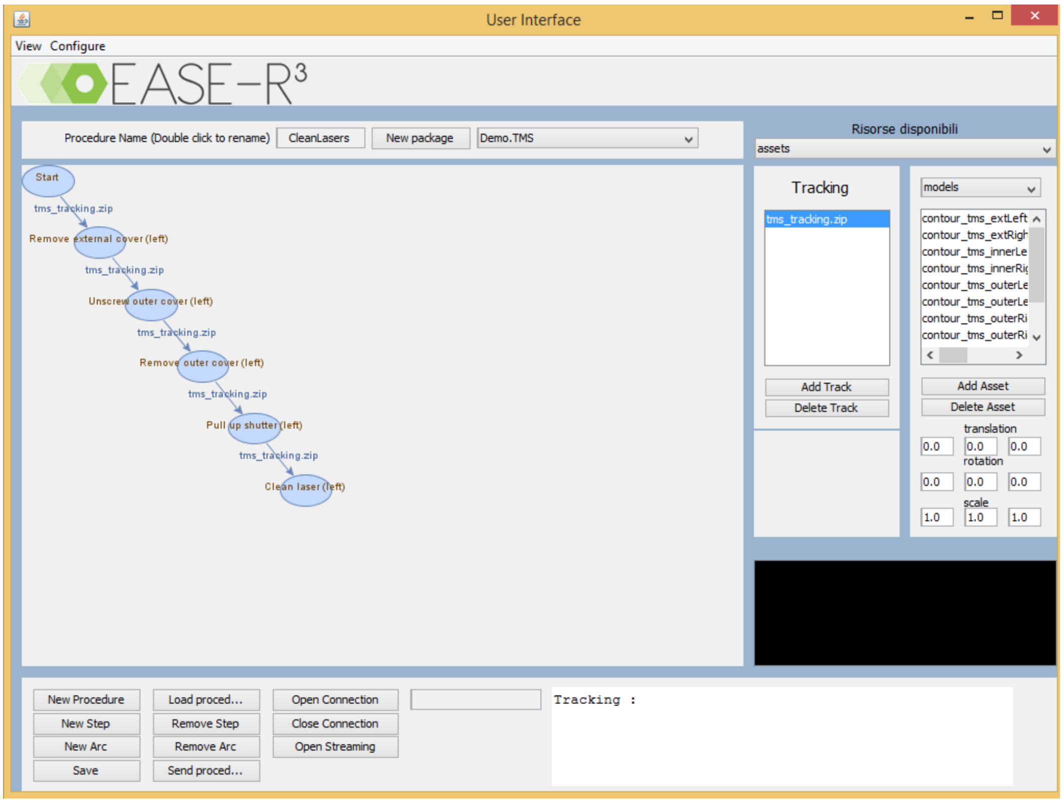

Fig. 3.17 The server user interface.

3.4.3 Server UI

Figure 3.17 shows the interface of the server application: the main section of the interface contains the state machine representation of the procedure. Each procedure consists of a series of edges and nodes. On the bottom-left corner a set of buttons allow the technician to define procedures and modify them with the available buttons:

- New Procedure: creates a new procedure: the first step is always named "Start";
- New Step: appends a new step to the current arc;
- New Arc: appends a new arc to the current step;
- Save: saves the current procedure;
- Load Procedure: loads a procedure previously saved;
- Remove Step: removes the selected step;
- Remove Arc: removes the selected arc.

The server application is equipped with a communication module that allows the technician at the headquarter to communicate with the maintainer and to see the video stream from the device camera; features of the connection between server and client are available through the following buttons:

- Send Procedure: sends the procedure to a remotely connected client application;
- Open Connection: enables the application to receive an incoming request from a client application;
- Close Connection: closes the connection channel;
- Open Streaming: starts the streaming from the camera of a remotely connected client.

The black box on the bottom right of the interface is a textual log window that provides information about the communication status with the client. The nodes represent the steps of the procedure to perform and they contain all the virtual aids, or assets, chosen by the technician. Edges represent the transitions from one state to another and are associated to a specific tracking configuration. A tracking configuration consists, in this case, of a CAD model, with a specific viewpoint and real world dimensions and it represents the real object the maintainer should interact with, e.g. a panel to open or a switch to turn on.

For each step the technician chooses the tracking configuration that better represents the view the maintainer should have of the real object to be managed: first, the step to be assigned the tracking configuration is selected. Then, the desired tracking configuration is chosen from the tracking column, the one on the right of the arc/node schema: a preview will be available under the “add track” and “delete track” buttons that can be used to add/delete a tracking configuration to the current step.

After that, the technician can add desired assets to the selected step of the procedure by selecting them from the rightmost column: a widget lists all the available assets and allows the technician to adjust scale, rotation and location of 3D models and animations aids in the virtual scene.

Selecting a step of the procedure, the white box at the bottom right of the interface will provide information on the current step:

- the name of the tracking configuration for the selected step;
- the list of the assets for the selected step.

How to Setup a Procedure

This section describes how to make an AR procedure; assets such as 3D models, animations and videos have to be already available. Necessary steps:

1. Press “New Procedure” and insert a name for the procedure. A “Start” step will be automatically created.
2. Press “New Arc” to add a new arc from the “Start” step.
3. Select the arc from the graphical representation.
4. Select a file from the Tracking column to see a preview of the tracking configuration. Navigate through the different configurations to choose the one to add

to the current arc. The tracking configuration should represent the first view of the object the maintainer should operate when starting the procedure step.

5. Press “Add Track” to associate the tracking configuration to the selected arc. The summary panel at the bottom right of the interface will be updated.

6. Press “New Step” to add a new step to the procedure and specify a name for it. The column right to the Tracking column lists all the possible assets the technician could add to each step.

7. Navigate through the available type of assets through the drop-down menu. Select an asset to add to the current step. If necessary (e.g. Animation on 3D Model), setup the translation/rotation/scale attributes to adapt the asset visualization compared to the tracking configuration. For 3D models/animations, assets have to be correctly placed in the scene space.

8. When ready press “Add asset”. The summary panel at the bottom right of the interface will be updated.

9. Repeat Steps 7-9 to add all the desired assets to the current step.

10. When ready to move on, it is possible to add a new arc and a new step to the procedure. A procedure is a set of sequential states.

11. Repeat Steps 2-10 to setup all the steps of the procedure.

12. When the procedure is ready, press “Save Procedure” to save it.

It is possible modify the procedure performing the following operations:

- Delete an arc or a step.

- Select an arc and change the tracking configuration, deleting the current one and adding another one.

- Select a step and remove or add assets. To remove an asset it is necessary to select it from the list and press “Delete Asset”.

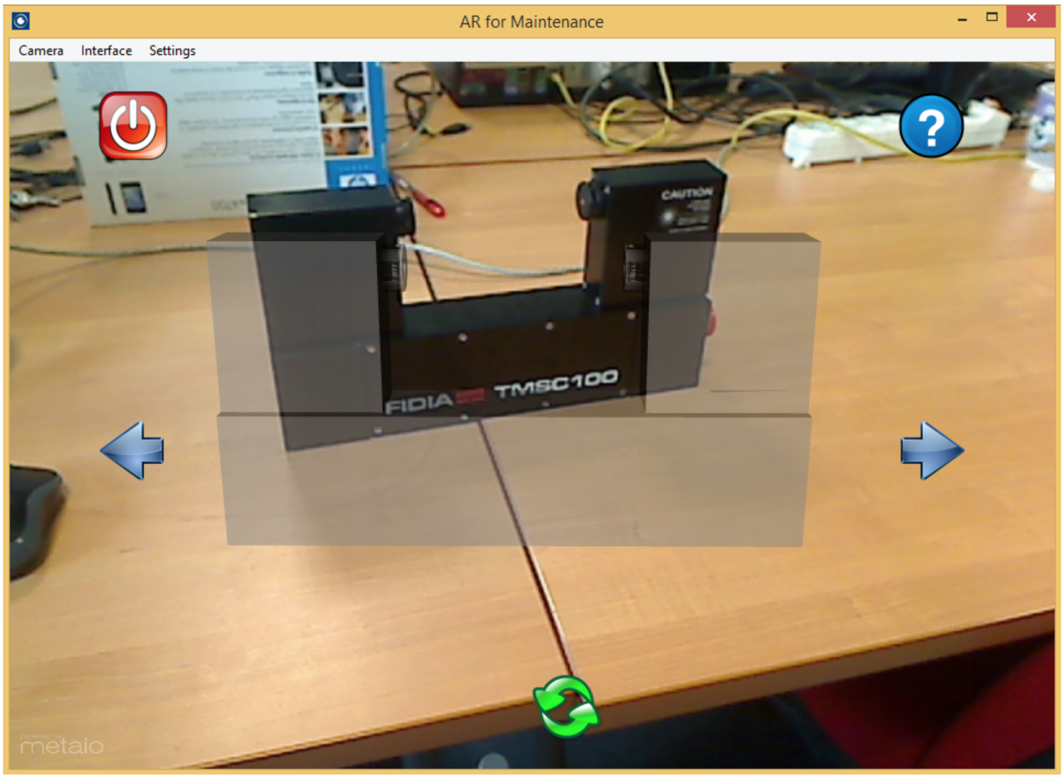

Fig. 3.18 The Windows client user interface.

3.4.4 Windows UI

Figure 3.18 shows the user interface for the windows client. The AR-glasses application comes with a speaking recognition module that maps all the commands available in the graphic interface for the Android application to vocal commands. In fact, the graphic interface is used only for debug purpose through the monitor of the computer connected to the AR-glasses, as the maintainer is supposed to use the vocal commands. The available commands are:

- ‘start procedure’ to start the practice;
- ‘next’ to move on to the following step of the procedure;
- ‘previous’ to go back to the previous step of the procedure;
- ‘reload’ or ‘restart’ to repeat the tracking recognition of the real object;
- ‘video’ to play, if available, the video asset for the current step;

- ‘instructions’ to repeat the audio asset for the current step;
- ‘assistance’ to open the communication channel with the instructor.
- ‘quit procedure’ to terminate the procedure.

It also lets users to turn on and off the speaking recognition module in order to avoid false positive recognition when they work in “noisy” environments or when they communicate with the technician (‘enable/disable speech’).

The bar positioned at the top of the interface allows the user to change some configuration options before starting the procedure with the AR-glasses. The “Camera Menu” presents a list of the available webcams. The maintainer can then select the AR-glasses camera and setup its preferences for future use of the application. The “refresh camera list” option updates the list of cameras connected to the computer. Only one camera can be selected to perform the procedure and it will be shown in the list. The “Interface Menu” presents the following options:

- See-through (enabled/disabled): if enabled the visual assets will be turned off;
- Stereo 3D (enabled/disabled): if enabled it will render the assets in stereo-mode for better visualization with the AR-glasses;
- Display buttons (enabled/disabled): if enabled it shows the debugging icons;
- Display user interface (enabled/disabled): if enabled the text assets will be displayed;
- Use speech recognition (enabled/disabled): if enabled the speech recognition module will be turned on;
- Full screen (enabled/disabled): show the interface at full screen resolution for better compatibility with the AR-glasses; disable to better adapt the application window in debug mode;
- Debug console (enable/disabled): if enabled, the log of the application will be shown.

The “Settings Menu” provides the following options:

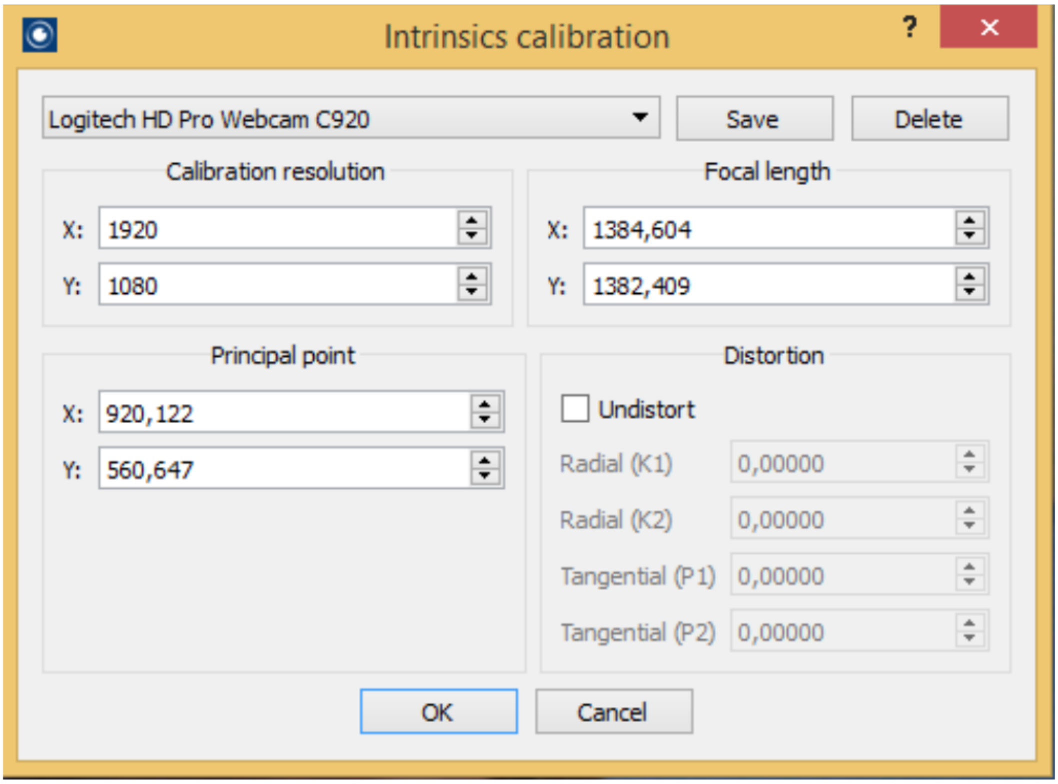

Fig. 3.19 The intrinsic calibration panel.

- Package directory: it allows the user to select the root folder where the package for each procedure is stored;
- Remote server: it allows the user to setup name and IP address for the Server application to request assistance;
- Camera intrinsic: it allows the user to modify the camera parameters;
- Hand-eye calibration: it allows the user to modify the AR-glasses parameters.

Figure 3.19 shows the "Intrinsic calibration" panel. This panel allows the user to setup the specific parameters of the camera (Video Resolution, Focal Length, Principal Point and Distortion) and save them. The drop down menu allows the user to select the available saved configuration for the 'known' cameras. The save button allows the user to save any change to the currently selected camera options and the delete option allows the user to discard a configuration. As a default, the currently active camera options will be shown.

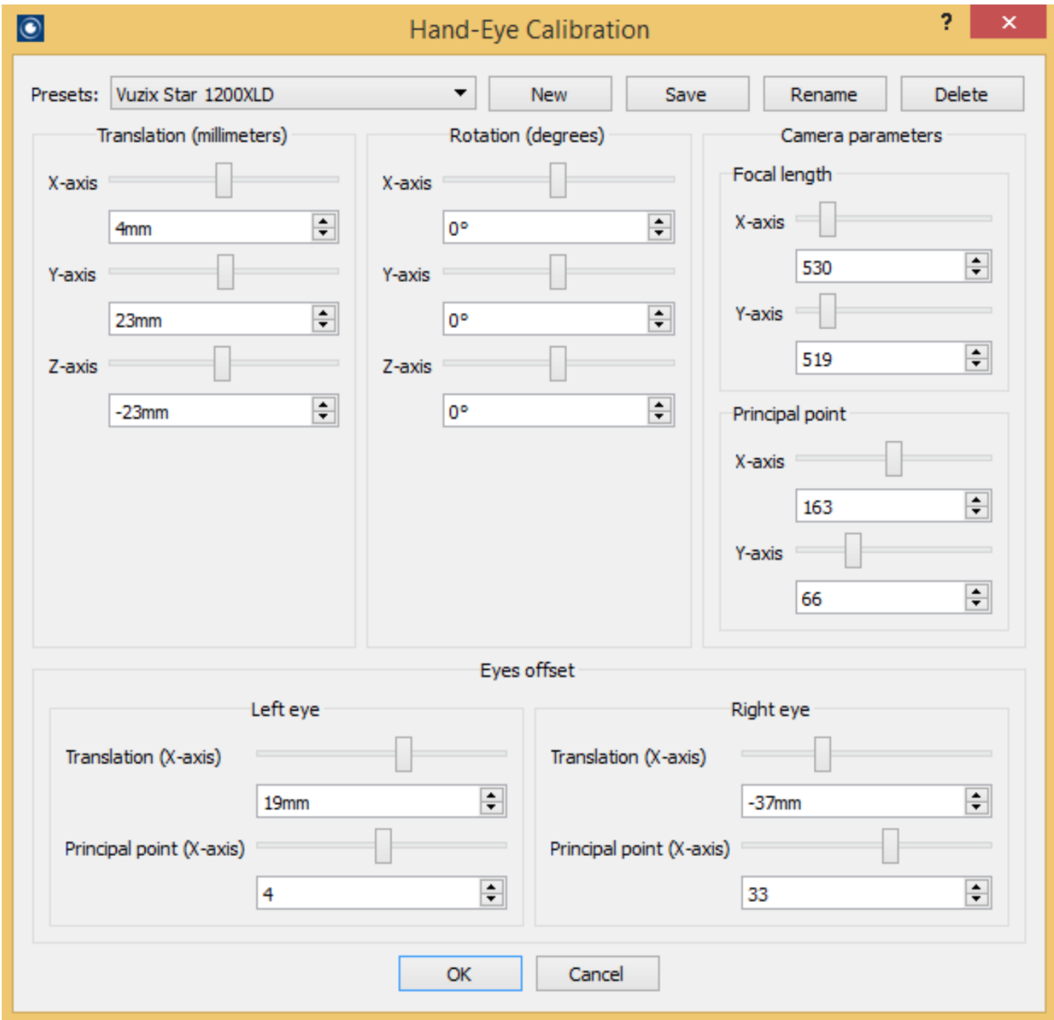

Fig. 3.20 The hand-eye calibration panel.

- Calibration resolution: the resolution used when the calibration was performed;
- Focal length: the camera's focal length;
- Principal point: the camera's principal point;
- Undistort: if selected, it is possible to modify the distortion parameters and the selected transformation will be applied.

Figure 3.20 shows the "Hand-Eye calibration" panel. This panel allows the user to modify the calibration parameters for the AR-glasses and save them as preset for future uses. The default preset presents all parameters with a values of 0, except for the focal length and the principal point that will be provided by Metaio. This panel allows for real time variation.

- ‘New’: creates a new preset (copying the values from the current one);
- ‘Save’: saves the preset;
- ‘Rename’: renames the preset;
- ‘Delete’: deletes the preset (if there are more than 1).

The translation, rotation and camera parameters represent the different parameters applied to the rendering camera. The Eyes offset parameters for the right and left eyes are the translation and principal point positions. These options are available only when the “Stereo 3D” option is enabled.

3.4.5 Android UI

Tablet

Figure 3.21 shows the user interface of the Android application. The two arrow icons on the two sides of the screen let the user to go back and forth through the steps of a procedure. The circle arrow at the bottom forces the tracking engine to repeat the recognition phase, e.g. to better align the virtual aids to the real object. If a video asset is available for the current step, a movie frame icon at the bottom right of the user interface allows the user to display the current step performed by an expert, thus outlining any difficult or ambiguous operations. The receiver icon at the bottom right allows the user to request a communication with the instructor for remote assistance.

Epson Moverio BT-200 Glasses

The interface of Moverio glasses is similar to the Windows client (Vuzix glasses). When the application is started, the list of procedures locally available is shown. Moverio glasses are connected to a touch-sensitive device, which allows the user to move a cursor and select a procedure. When a procedure is selected, steps are exactly the same as the ones described for the Vuzix AR glasses: a silhouette has to be aligned with the real object, and when the Metaio application is tracking

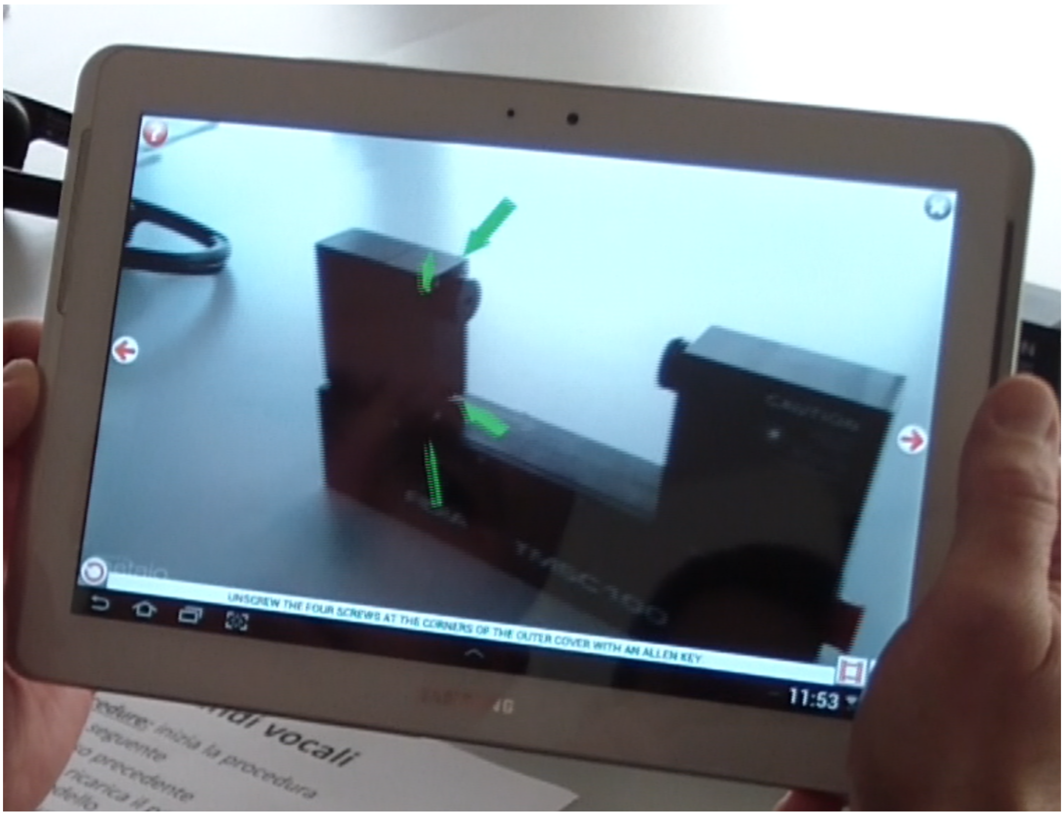

Fig. 3.21 The Android client user interface.

the object, a set of assets is displayed. The format of procedures is the same as the one used by the Windows client. Inputs are provided by a speech interface. Two different solutions have been investigated: the first one runs on the Android device the speech recognition engine, whereas the second solution uses an external PC (connected by Bluetooth to the microphone) to perform the speech recognition task. The first solution provides real mobility but the available library is available only for the English language. As performance of speech recognition engines are strongly dependent on speakers, and as the two selected use cases involve Italian companies (and technicians) an “external” solution well supporting Italian mother tongue speakers has been considered. The visualization of 3D assets is performed (as for Vuzix glasses) in a stereoscopic way; this allows the user to correctly align computer generated 3D models with real objects but also involve some issues concerning the visualization of GUI widgets. As icons and 3D assets are, in general, displayed on different “focal planes” it is very difficult for the user to correctly visualize concurrently assets and the interface. Therefore, the application

Fig. 3.22 Hiding the assets.

allows the user to hide the interface when assets are displayed. An example is shown in Figure 3.22 where assets of a step of the ball bearing disassembly procedure are shown.

3.5 Speech Interfaces for AR-base applications

Associating an icon with one or more voice commands is a rather common practice on desktop computers, where accessibility features are available in major operating systems. A similar consideration applies to the mobile world, where popular speech recognition-based personal assistants let the users control their devices using voice commands. However, although voice commands (with some exceptions) can be exploited in principle to control any application functionalities, their use is often limited to native applications, because third- party developers need to code all the interactions explicitly [196]. Addressing the above limitations, several approaches have been developed to automatically create a speech interface by interpreting what is displayed on screen [197], [196].

3.5.1 Automatic Interface Generation Techniques

Automatic generation of UIs is pursued not just for accessibility purposes or only in the context of speech interfaces. In fact, the focus has been more on graphic UIs, with the aim of reducing development costs [198], enhancing usability [199], maximizing performance [200], supporting personalization [201] and fostering portability on heterogeneous devices [202]. Although, in most cases, the techniques for automatic interface generation have been designed to arrange graphic elements in a suitable layout or to let the user choose the best widget for a given interaction task, there are several works operating at the level of individual graphic elements and, more specifically, of icons. Icons play, today, a key role in the graphic interface, as they can convey, in a condensed form, the meanings of the functionalities they represent. Icons are especially important for novice users who use interactive systems only infrequently and are assumed to be capable of transcending language barriers [203]. However, drawing or selecting icons for an interface is not a trivial task. Ideally, icons should be capable of activating proper mental models in the users, but this is not always the case, mainly because of the phenomenon known as icon "ambiguity": different individuals may interpret the same icon in different ways, thus associating it with different functionalities. In fact, icon interpretation depends both on the actual user's mental model and on other factors, such as context, culture and experience. A common approach adopted in selecting or drawing icons is to ensure that the "articolatory distance" between the physical form and the implied meaning is minimal [204]. For instance, the icon that shows a person discarding trash will have a small articolatory distance, because the iconic object is self- explanatory. On the contrary, for an icon with a power button, the distance has to be much higher, because the graphic symbol is rather abstract and its effect on the controlled application needs to be learned or discovered. The steps that need to be taken for minimizing this distance and for making the criteria for an icon to be effective, include, among others, legibility, distinctiveness, comprehension, memorability, familiarity, and reaction time [205, 206]. The strategies proposed for assessing icons date back to 1960's and include appropriateness, preference rating, naming and matching, comprehension, recognition and recall as well as intuitiveness tests [207–209]. Given the complexity of design and selection tasks, the research efforts have been devoted to finding a way to automate the process. For instance, Morioka et al. propose an approach to automatically generate complex icons by combining basic graphic patterns [210]. Keogh et al. develops a methodology for replacing the icons of a file system with automatically- created icons that reflect the actual content of files [211]. Oda and Itoh propose a somewhat similar strategy to automatically generate icons for music files by matching tunes with images [212]. Setlur and Mackinlay designs a method to automatically generate large icon libraries for use in the visualization of categorical

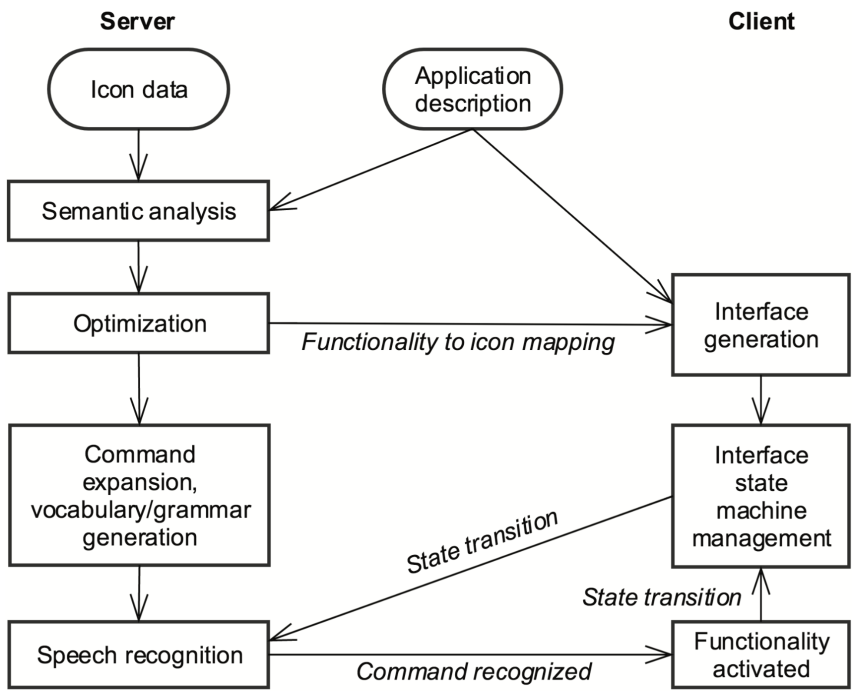

Fig. 3.23 The IIF architecture.

data [213], whereas Pickering addresses the problem of icon selection, with the aim of enforcing consistency in the usage of graphic signs within, as well as across, applications, and enhancing text expressiveness [214].

3.5.2 Intuitive Interface Framework

The Intuitive Interface Framework (IIF) was structured on the lines of client/server architecture, as shown in Figure 3.23. The server is responsible for generating the target application interface, including the vocabulary and grammar for the speech recognizer, as well as the corresponding icon-based visual cues. This offline process can be performed only once per each application. At run-time, the server manages the speech recognition engine and communicates to the client the functionalities to be activated, based on the commands issued. The client reacts by activating the specific functionality and notifying the server about interface changes. The design strategy adopted here significantly reduces the dependence on the client, thus allowing the target application to be implemented using different technologies and/or programming languages.

Server

The off-line interface generation process, on the server side, receives as input a description of functionalities of the target application together with data about the available icons. Its output is a mapping that associates each icon (to be displayed on the application interface) to a specific functionality and a set of implied speech commands (each expected to be evoked by the corresponding icon). Application functionalities are described by the developer, using short phrases of one or more words. Icon data comes from an annotated catalog, which was created by interviewing generic subjects and collecting short descriptions of their possible meanings. Application and icon descriptions are semantically analyzed to measure the correlation between functionalities and icons (defining the extent to which their meanings are related), as well as the distance among icons (defining the extent to which their meanings are unrelated). The mapping is determined by an optimization method, which considers the outcome of the semantic analysis to maximize interface flexibility and robustness, while preserving visual consistency. Finally, voice commands are generated by lexically and semantically expanding phrases associated with the functionalities and icons included in the mapping. In the following pages, the modules and methodologies developed for realizing the above process are described in detail.

The developer is expected to provide a description of all client application functionalities using a state machine model. The state machine was assumed to represent the flow of application interface, indicating which functionalities would be available at any given moment and how they affect the interface when activated. Figure 3.24 is a graphical representation of the state machine for the application considered in the case study.

The state machine is described using an extension of the W3C's State Chart XML (SCXML) notation [215]. Figure 3.25 is an excerpt of the SCXML description that could be provided by the developer for the considered application. Functionalities are described as events (attribute event) that activate transitions (tag <transition>) between states (tag <state>). For each functionality, one or more phrases are provided using the <phrase> tag. Moreover, the developer may specify relationships among functionalities by defining the so-called affinity groups with the <affinity-group> tag. For instance, "next step" and "previous step" in Figure 3.24 are two affine functionalities, as they are complementary. Similarly, "play video", "pause video" and "stop video" are affine, because they are all related to video control.

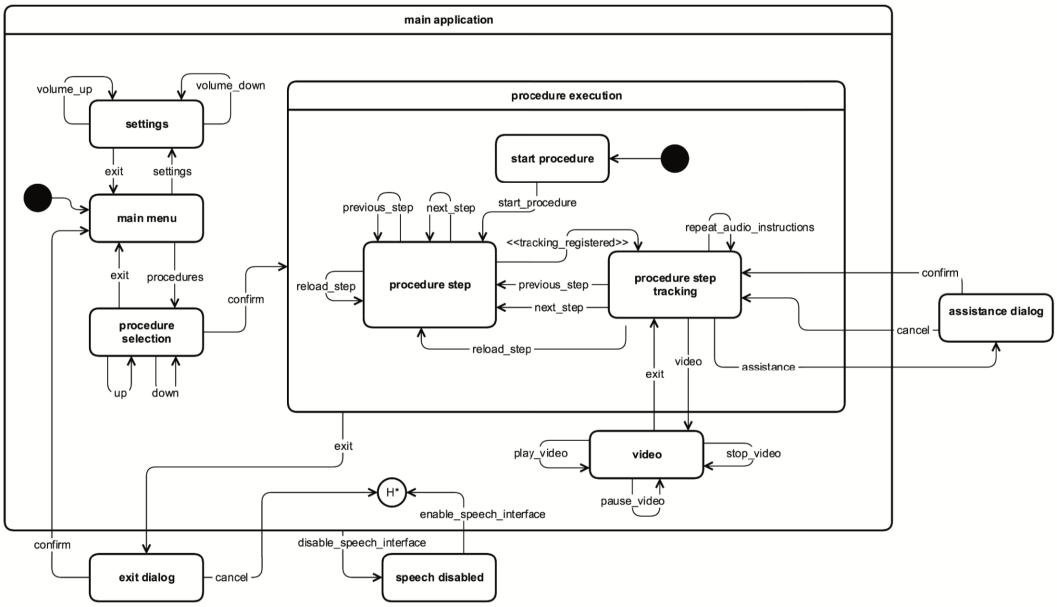

Fig. 3.24 State chart representing the interface of the application for AR maintenance considered in the used case.

```
<scxml initial="main menu" app="ARMaintenance">
  <state id="procedure step tracking">
    <transition event="previous_step" target="procedure step"/>
    <transition event="next_step" target="procedure step"/>
    <transition event="assistance" target="assistance dialog"/>
    <transition event="video" target="video"/>
    <transition event="exit" target="exit dialog"/>
      ...
  </state>
  ...
  <event name="next_step">
    <phrase value="avanti"/>
    <phrase value="passo successivo"/>
  </event>
  <event value="assistance"/>
    <phrase value="chiedi aiuto"/>
  </event>
  ...
  <affinity-group type="move state">
    <event value="previous_step"/>
    <event value="next_step"/>
  </affinity-group>
</scxml>
```

Fig. 3.25 Developer-provided state machine for the user interface showing the functionalities (and text-based descriptions) and relations among theme.

Icon Data Collection: The proposed methodology requires a catalog of icons, annotated with implied functionalities. The catalog should, in principle, be sufficiently comprehensive to include all the icons required to create the interfaces of various applications. Indeed, a number of icon sets exist in which each icon is assigned one or more significant names. However, what the proposed method needs is a collection of meanings, each expressed as a phrase of one or more words, including frequency statistics. Such an annotated catalog was created by using approaches that are rather common in usability tests of graphic interfaces [209], where one or more sets of alternative visual signs are evaluated through questionnaires and user studies. Instead of having the icons drawn from scratch by a graphic designer or assembling them from multiple sources on the Web, a publicly-available set named Clear Icons [216], comprising 50 general-purpose icons, was used. Then, an online questionnaire was designed, presenting the icons in a random order. Participants (selected among university students) were asked to provide, for each icon, one or more phrases describing the functionality that could be possibly activated in a generic device, application or appliance. The questionnaire was returned by 28 subjects and each icon received, on average, 41 descriptions[1].

Semantic Analysis: The process of semantic analysis exploits information from the multiWordNet 1.5.0 lexical database [217], an Italian version of the well-known WordNet thesaurus [218]. As has been mentioned, the goal was to calculate the functionality for icon correlation and the icon to icon distance measure for exploitation in the optimization module. Phrases associated to every icon and functionality were processed to remove stop words. Then, the remaining words were linked to lemmas in the lexical database by applying stemming and lemmatization mechanisms. For each word, the part of speech was recorded, together with the details about how the lemma was obtained (e.g., word found as is, conjugated verb, etc.). With such information, correlation w_{ij}^{corr} between icon g_i and functionality f_j was computed as follows. First, considering the lemmas, all the synonyms and antonyms were extracted. Then, a similarity index $sim(p_s, p_t)$ for two generic phrases p_s and p_t composed of N_s^W and N_t^W words was defined by considering the number of matching lemmas (n_{st}^L), synonyms (n_{st}^S) and antonyms (n_{st}^A), for each word in the two phrases. If $n_{st}^A > 0$, then $sim(p_s, p_t) = 0$. When $n_{st}^A = 0$ and $n_{st}^L = N_s^W$, then $sim(p_s, p_t) = 1$. Otherwise, if $min(n_{st}^L + n_{st}^S, N_s^W) = N_s^W$, then $sim(p_s, p_t) = \frac{(n_{st}^L + \frac{1}{2} n_{st}^S)}{(n_{st}^L + n_{st}^S)}$; if $n_{st}^L + n_{st}^S \neq 0$, then $sim(p_s, p_t) = \frac{(n_{st}^L + \frac{3}{4} n_{st}^S)}{[(n_{st}^L + n_{st}^S) max(N_s^W, N_t^W)]}$. The weights in the above equations were empirically defined to account for the fact that synonyms could include meanings that are far from those of the original word. Moreover, considering the fact that the number of words in the two sentences could be different, they

[1]Results available at http://intelligenthci.altervista.org/questionario/

also took into account the possible meaning of missing information. Finally, the correlation was obtained as

$$w_{ij}^{corr} = \frac{1}{N_i^P N_j^P} \sum_{u=1}^{N_i^P} \sum_{v=1}^{N_j^P} sim(p_u^i, p_v^j) \tag{3.1}$$

where p_u^i is the u-th phrase of icon g_i, whereas p_v^j is the v-th phrase of functionality f_j . N_i^P and N_j^P are the total number of phrases associated to icon g_i and functionality f_j , respectively. Distance $w_{i_1 i_2}^{dist}$ between icons g_{i_1} and g_{i_2} was computed similarly, as 1 minus the icon-to-icon correlation value.

Optimization: The mapping process between icons and functionalities was formulated as an optimization problem. To define the problem, application description was considered as having been composed by N^S states and N^F functionalities. By defining N^G as the overall number of available icons, a binary decision variable $x_{ij} \in [0,1]$ was introduced so that

$$x_{ij} \begin{cases} 1, \text{ if icon } g_i \text{ is associated to functionality } f_j \\ 0, \text{ otherwise} \end{cases} \tag{3.2}$$

The following constraints were enforced

$$\begin{aligned} &\sum_{i=1}^{N^G} x_{ij} = 1 \; \forall_j = 1,2,...,N^F \\ &\sum_{j=1}^{N^F} x_{ij} \leq 1 \; \forall_i = 1,2,...,N^G \end{aligned} \tag{3.3}$$

ensuring that each functionality was mapped exactly to one icon, and the same icon was not associated to more than one functionality. Two helper functions were also defined as

$$\begin{aligned} &q_{sol}(i) = \sum_{j=1}^{N^F} x_{ij} \\ &q_{state}(i,s) = \sum_{j \in j | a_j \in S_s} x_{ij} \end{aligned} \tag{3.4}$$

where $q_{sol}(i) = 1$ if icon g_i is in the solution (i.e., associated to a functionality) and $q_{state}(i,s) = 1$ if icon g_i is present in state S_s (i.e., mapped to a functionality in that state). The solution needs to maximize the correlation between functionality and associated icon, the icon-to-icon distance in a given state, and the so-called icon affinity (defined later in this section). The metrics for correlation and distance were derived in the semantic analysis step. Cost functions for correlation and distance are defined as

$$\begin{aligned} obj_{corr} &= max \sum_{i=1}^{N^G} \sum_{j=1}^{N^F} w_{ij}^{corr} x_{ij} \\ obj_{dist} &= max \sum_{s=1}^{N^S} \sum_{i_1=1}^{N^G} \sum_{i_2=2, i_1 \neq i_2}^{N^G} w_{ij}^{dist} q_{state}(i_1, s) q_{state}(i_2, s) \end{aligned} \tag{3.5}$$

Overall correlation is the sum of the w_{ij}^{corr} coefficients of all the functionality-icon pairs. Distance was calculated for every state, considering the distance coefficient $w_{i_1 i_2}^{dist}$ for each pair of icons g_{i_1} and g_{i_2} appearing in that state. Icon affinity is related to the visual aspect of icons. In fact, the icons' styles should be mutually consistent, especially when they are associated to related functionalities. For example, if the functionality "next step" is associated to a "right arrow" icon, then the "previous step" functionality, if present, should be associated to a "left arrow" icon with the same style. As for functionalities, icons can be associated to one or more affinity groups. Affinity $w_{i_1 i_2}^{aff}$ between two icons g_{i_1} and g_{i_2} is calculated as the ratio between the number of affinity groups shared and the total number of groups defined for icons. By referring to F_i^{aff} as the i-th functionality affinity group and to N^A as the total number of groups specified for functionalities, the following cost functions can be defined

$$\begin{aligned} obj_{aff}^{overall} &= max \sum_{i=1}^{N^G} \sum_{i_2=2, i_1 \neq i_2}^{N^G} w_{i_1 i_2}^{aff} q_{sol}(i_q) q_{sol}(i_2) \\ obj_{aff} &= max \sum_{k=1}^{N^A} \sum_{i_1=1}^{N^G} \sum_{i_2=2, i_1 \neq i_2}^{N^G} sum_{j_1, j_2 \in j | f_j \in F_k^{aff}, j_1 \neq j_2} w_{i_1 i_2}^{aff} x_{i_1 j_1} x_{i_2 j_2} \end{aligned} \tag{3.6}$$

The first function represents the affinity of all icons belonging to the solution, and the second one the affinity among icons associated to functionalities of a particular affinity group. Although the overall affinity is already being maximized by the first function, the second one helps to speed up algorithm convergence by avoiding icons that do not match with affine functionalities.

The problem was solved by using a multi-objective meta- heuristic optimization approach to provide scalability in terms of number of available icons and number of functionalities, because solution space has a size of $N^G!/(N^G - N^F)!$. Implementation was created by using the jMetal [219] framework. The solution was represented by an array of integers, indicating icon indexes, whereas position in the array is associated to the functionality. The NSGA-II genetic algorithm was used with a polynomial mutation and a SBX crossover operator. The algorithm can obtain high-quality solutions, though convergence speed can still be improved, e.g., by adopting a different strategy for representing the solution.

Command Expansion: Once the optimal mapping was found, the last step in the generation of the interface consists in building the vocabulary and grammar to handle commands by the speech recognition engine. As conflicts can arise among commands, they were confined to a single state, and expansion was performed separately for each state. For each state and for each functionality-icon pair in that state, the algorithm considered all the phrases associated to both functionalities and icons. Lemmas linked to each word were expanded into synonyms and the verbs were conjugated to imperative form. Expanded terms were combined into new phrases by computing meaningful combinations. Phrases were weighted to account for their frequency and length by considering the way they were generated. Finally, the phrases were converted into commands by replacing some words (e.g., articles such as “a”, “the”, etc.) with wildcards that match any word the user pronounces, thus increasing flexibility during recognition. Once all the functionality-icon pairs in a state were processed, the associated commands were filtered, based on their weight. Potential overlaps among commands were solved so that, at the end of the process, each functionality would be associated to an independent set of commands.

Speech Recognition: This module was developed using the Microsoft Speech Platform13, which presently supports 26 languages. Implementation for this study operated in Italian language to limit recognition errors possibly caused by mis-pronunciation.

Fig. 3.26 A screenshot of the client application (obtained by disabling see-through mode).

Client

The client side of the system was implemented by a library that manages the user interface and the communications with the server. The target application adopted in the considered case [61] was developed using the Metaio SDK [180] AR framework. Specifically, it was built using the AREL technology, which lets the user create the interface as an HTML page with the logic defined in JavaScript. Hence, the client library was implemented in Javascript. Nonetheless, it can be easily implemented by using different programming languages and in different environments, thus allowing the deployment of the devised interface generation mechanism on heterogeneous platforms. Initially, the application configures the client library with the state machine of its interface (see Figure 3.24, for the case selected to use). It also sets up hooks between interface functionalities and the codes that actually implement them. Once the application is started, the Javascript library connects to the server, requests the functionality to icon mapping, and creates the user interface. Finally, it moves to the first state of the interface (displaying the corresponding icons) and notifies the server.

When a state update is received, the server loads the associated vocabulary and grammar and starts the speech recognition engine. Every time a command is recognized with enough confidence, the client is informed of invoking the matching functionality, triggering its activation. Figure 3.26 depicts a screenshot of the client application (obtained by disabling see-through mode). Maintenance operations are carried out in stereoscopic mode, which allows the user to focus both on the object to be maintained and the 3D assets used as AR hints. However, it was discovered that most users find it difficult to focus the icons shown in that mode, because they lie on a virtual plane at distances different than those of both virtual and real contents. Therefore, by adopting the same approach as the one pursued for native icons of the Moverio glasses, stereoscopy was disabled for icons. Icons are rendered on the sides, in such a way that, for instance, the icons on the left can be seen only with the left eye.

3.6 AR Interface Robustness

One switch (or single switch) interfaces have been deeply investigated in order to support the interface design for people with different kinds of disability. A traditional interface provides a direct selection paradigm, thus enabling users to activate any available command; in other words, a sort of "random access command" modality is supported. This kind of access requires a great level of interaction between user and machine, which is not available for people with severe motor or cerebral disabilities. One switch interfaces try to overcome this issue by presenting available commands in a sequential way: the user can activate a desired command by pressing a button, by a vocal input, by blinking, or by any sort of input that can be assimilated to a switch activation.

It is immediately clear how the scanning algorithm, which presents sequentially available commands, is a key issue for the interface usability; in particular, the scanning latency (also called scanning delay) has to be accurately tuned. Different scanning algorithms can be implemented and they are categorized as [220]:

1. regular or automatic - selectable elements are scanned cyclically and the user selects the desired command when highlighted;
2. inverse - the scanning selection advances only when the "switch" is continuously activated and the user can select a command by releasing the switch on it;
3. step - successive (discrete) switch triggers allow the user to select the desired command.

One switch selection interfaces are used in different applications, usually to improve the daily life of impaired people, ranging from text entry [221] to video games [222]. For instance, an adaptive scanning algorithm is proposed in [223] to efficiently perform text entry tasks, whereas a robotic arm is controlled by a single switch user interface to support people with less muscular strength in [224]. Wheelchairs can be driven by single switch interfaces [225] and a single switch scanning interface is used in [226] to allow people with amyotrophic lateral sclerosis to control a keyboard by eye control. Mouse manipulation has been implemented by a single switch solution in [227], whereas more complex human-computer interactions are proposed in [228] and [229]; in [228] objects on the screen are clustered by a sort of quad-tree algorithm in order to speed-up their selection, whereas a scanning keyboard has been implemented in [229] to allow children who are severely physically disabled to access microcomputers for writing, playing, and engaging in educational activities. Internet navigation is also available: a web browser controlled by a single switch interface has been presented in [230]. Although all these solutions based on scanning algorithms are considered slower than direct-selection applications, they usually provide a more robust interaction.

The idea described in this section is to use a speech-based interface to trigger a scanning selection; both virtual and augmented reality applications, where other input modalities are prevented, can take advantage of the proposed solution. For instance, wearable AR-based applications for maintenance allow technicians to perform hands-free tasks [231] and neither touch-based nor touch-less input paradigms can be used. On the other hand, speech recognition, considered natural and intuitive, might be not robust enough to be used in noisy environments. For this reason a sort of hybrid input speech interface is proposed: commands are not directly selectable but are sequentially activable by a scanning algorithm. An automatic scanning and an inverse scanning algorithm controlled by a single word have been implemented. Moreover, as a vocal switch is not necessarily binary as a physical one, a bidirectional scanning algorithm is also presented. In this case, two words are used to scan the command list and a third word is used to select the desired command. The goal of this work is to compare possible one switch scanning interfaces with a traditional multiword speech-based solution and obtain some indications about usability in noisy environments. The target application developed for the considered use case [61] is an AR application for performing maintenance procedures. The use case requires the user to complete a sequence of steps in order to accomplish a maintenance

Fig. 3.27 A screenshot of the application user interface.

task and the application provides assistance to him/her through AR contents. Since each step requires the user to perform operations that potentially involve the use of both hands, such as unscrewing bolts or removing components from a machinery, a hands-free interface is necessary. At each step of the procedure, a set of icons is used to inform the user on the available functionality provided by the application, as showed in Figure 3.27. Wearable devices present some limitations to the deploying of an application that involves tracking algorithms, graphic resources for displaying AR contents and a speech recognition system to provide the user interaction. The computing power may be inadequate to process smoothly all the required resources and the high computing tasks may dramatically reduce the battery life of the device, thus making it impossible to complete the given task. Moreover, the libraries for speech recognition available for wearable devices are not flexible and responsive as the ones available for desktop environment, especially in terms of robustness and languages supported. For these reasons, the speech recognition system has been developed on a desktop system and it works as the server side of the proposed solution, returning the recognized commands to the AR application running on the wearable device.

The application's framework consists of a client-server architecture, as showed in Figure 3.28. The user pronounces commands into a Bluetooth microphone directly connected to the server. The direct connection between the microphone and the server allows to reduce the computational power required by the client device. The server

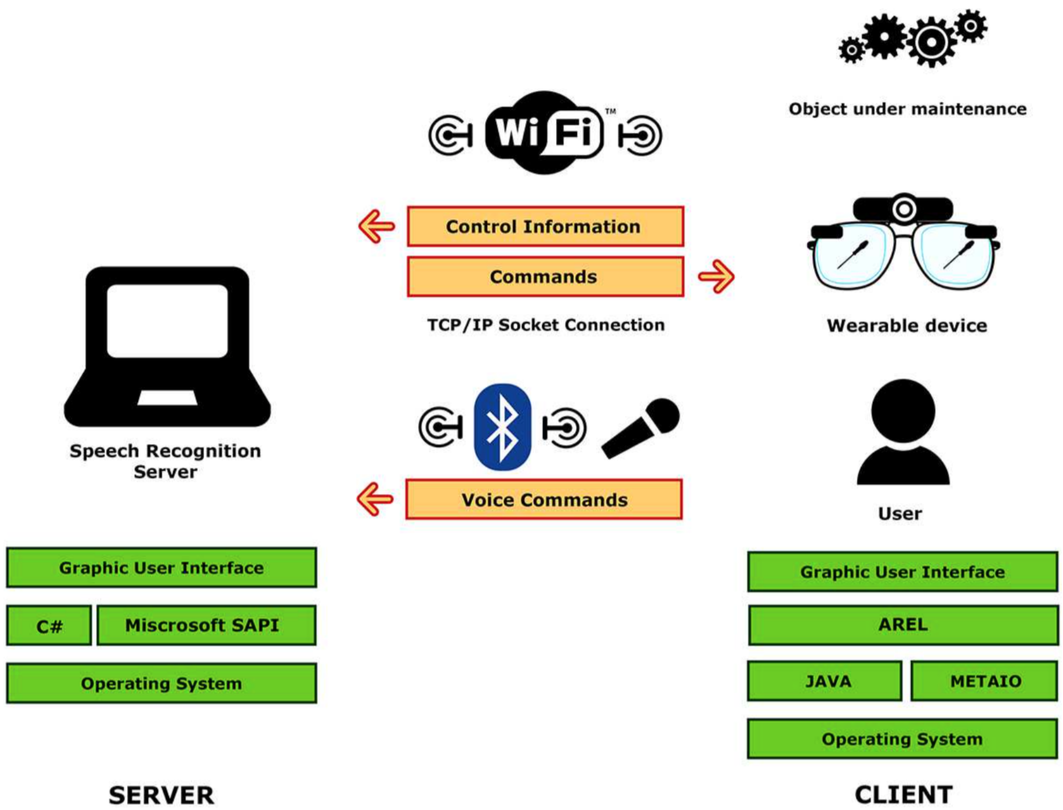

Fig. 3.28 Architecture of the proposed application.

handles the speech recognition and communicates to the client the functionality to be activated based on the uttered command. Then the client activates the complementary functionality and, if a change to the client interface occurs, it notifies it to the server in order to load the corresponding set of commands. The communication between client and server occurs on a local Wi-Fi network, through a socket connection. The client side of the system consists of an AR application for AR-Glasses that manages the user interface and the communications with the server. The AREL technology has been adopted to build the application, using the Metaio SDK.

First of all, the server loads a state machine representation of the client user interface, provided in an extension of the SCXML notation as presented in [60]. The state machine is expected to describe the layout of the client interface, as each state lists the available functionalities and how they modify the interface current state when activated. Moreover, the specific set of words to be recognized and mapped to a specific functionality is defined. Whenever a functionality compels the UI to change to another state, the server is notified and the corresponding set of words to be recognized is loaded. The speech recognition module has been developed using the Microsoft Speech Platform [232], which supports 26 different languages. The current implementation of the system operates in the Italian language, for the sake of reducing recognition errors caused by mispronunciation. At the beginning, the application asks users for choosing among the four different available interfaces: multi-word speech recognition-based system (MW), a one switch interface based on an automatic scanning algorithm (AVOS), a one switch interface based on an inverse scanning algorithm (IVOS) and a one switch interface based on a bidirectional scanning algorithm (BVOS). Then, the JavaScript library connects to the server and creates the user interface for the first state, displaying the corresponding icons, and finally it notifies the server. When a state update for the UI is received, the server loads the corresponding vocabulary and grammar. Every time a command is recognized with enough confidence, the client is notified that the matching functionality is invoked, thus activating it. Moreover, as the server provides information on the level of confidence when recognizing a command, a colored rectangle is shown in the top right corner of the UI, in order to provide a visual feedback to the user actions. The rectangle will assume three different colors, depending on the degree of confidence in the recognition phase:

1. green if the command was correctly recognized;
2. yellow if the command was present in the current set of recognizable words but the level of confidence was too low;
3. red if the pronounced command was completely unintelligible.

Moreover, each time an icon is activated, the background is set to transparent and a blue border appears for 500 ms to give a visual feedback to the user. The icons corresponding to the available functionalities are displayed on the left and right side of the interface, in a circular buffer: when the last icon is reached, the following one will be the first one, and vice versa. The four different interfaces provide the following interaction systems:

1. MW. In this modality, when a command is correctly recognized, the corresponding icon is highlighted with a blue border.
2. AVOS. In this case, the icons are highlighted one at a time with a green background and a latency of 4500 ms and if the user pronounces the confirmation

command, the currently highlighted icon is activated. After 3000 ms, the icon background shades to red to advise the user that it is too late to start pronouncing the command, as the following icon would be activated. Finally, the background is turned off for the remaining 500 ms, before activating the following icon. The total latency was determined after numerous tests to provide the best trade-off between the waiting time for the user (to be minimized) and the time necessary to pronounce the word, process it on the server and provide a feedback to the client if correctly recognized (in a reasonable time).

3. IVOS. In this case, when a new state of the UI is loaded, the first icon is highlighted with a green background. When the user pronounces the command to advance, the background of the current icon is turned off, and the next one is highlighted. If the user highlights an icon and then he/she does not pronounce the command for 4500 ms, the action corresponding to the current icon is activated. After 3000 ms, the icon background shades to red to advise the user that it is too late to start pronouncing the command, as the current icon is going to be activated. Finally, the background is turned off and a blue border appears to advise the user that the current icon is being activated. The total latency time was determined in the same way as for AVOS.

4. BVOS. In this case, when a new state of the UI is loaded, the first icon is highlighted with a green background. The user can then activate the current icon with the confirmation command, or move to the previous or next icon with the specific command, thus moving the highlighting to another icon.

3.7 Conclusions

This chapter illustrates the steps that were performed in order to design and develop the proposed frameworks and solutions. After selecting the tools for the job, a workflow has been proposed in order to assess the development of AR solutions. The adoption of the proposed workflow has led to the development of a complete Augmented Maintenance Framework, that could be used by maintainers for both training and real maintenance procedures. Moreover, further developments have focused on the user interface, trying to address both the easiness of use and the robustness of the interaction. Overall, the proposed framework provides the following functionalities:

- the instructor can create maintenance procedures, adding different kinds of assets such as text labels, images, animations, videos and audio tracks;
- the proposed tracking solution is independent from environmental conditions and it does not require the usage of graphical artifacts such as QR codes;
- the client allows the user to navigate through the steps of the procedure;
- the client allows the user to remotely connect to a server to request further instructions from a senior technician and/or to request an updated or modified version of the current procedure that better fit the real situation;
- the proposed client-server architecture could be used as a remote training system;
- the user interface is customizable, depending on user preferences;
- the client allows the user to choose the procedure from a list of available ones;
- the client allows the user to choose the language;
- the speech recognition interface could be tuned depending on the environmental condition.

Moreover, as detailed in Figure 3.12, all the challenges depicted in the previous chapter have been addressed through the proposed framework. Among the list of user requirements defined in section 2.6, the frameworks and solutions proposed in this chapter address most of the issues listed in section 2.7, even if further improvements are possible. For example, even if the AMF provides specific aids depending on the scene framed by the device, it does not recognize autonomously if the current step has been completed or not. However, chapter 5 will detail some of the possible future advancements that could be researched. The next chapter will illustrate the tests designed and performed to assess the proposed solutions.

Chapter 4

Tests and Results

Part of the work described in this chapter has also been previously published in [3] and [4], [60], [61] and [62]

This chapter illustrates the tests that have been performed to evaluate the proposed frameworks. Each section also presents the assessment of the obtained results for that specific set of tests. Section 4.1 describes the preliminary tests, related to the workflow analysis proposed in 3.2. The following section describes the tests used to evaluate the AR procedures' framework described in 3.3. Section 4.3 focuses on the tests related to the assessment of the IIF described in 3.4.2. Afterwards, section 4.4 illustrates the tests proposed for evaluating the robustness of the speech interfaces presented in 3.5.

4.1 Preliminary Tests

The case study described in section 3.3.4 has been tested by using the real object (the ink-jet printer) in order to assess the efficiency of the proposed tracking systems. Tests on the real object have been repeated using 3D CAD tracking and 2D tracking. The second solution has been tested with both renderings and photos of the real object. As one of the main focus of the research is that the procedure should be repeatable independently of the environmental conditions, the photos were taken through the tracking camera and edited to remove all environmental information. As detailed before, since the last three steps of the procedure (5,6 and 7) are specular to the first three ones, only steps 1-4 were taken into account for these tests.

Four parameters have been considered in each test, in order to evaluate the performance of the two solutions: tracking object quality, recognition threshold, luminosity, alignment precision. The tracking object quality defines the quality of the image or CAD model for the tracking algorithm. Both systems have their own rules to finely craft the tracking object. Metaio Creator rates with a three star system how much the chosen tracking object is suitable. The better the rating, the easier for the system to recognize the object and avoid false positives, providing a robust solution. Recognition threshold evaluates the similarity parameter used to calibrate the matching between the tracking virtual object and the real one ranging from 0.00 to 1.00. Metaio suggests the range 0.30 - 0.70 for better results, as values below 0.30 will lead to a too inaccurate recognition and values over 0.7 could make it difficult to recognize the object. A low value means that the system is less robust: false positives or alignment errors of the assets might occur in this case. Luminosity has been evaluated to test the robustness of the systems in different situations. The tests have been performed with low light (30 lux), medium light (160 lux) and high light (300 lux), where low light represents a nearly dark room and high light represents a well illuminated office desk. The luminosity value is approximated and it is evaluated through the luminosity sensor of a Nexus 10" Tablet. 3D CAD recognition should always work, as it does not rely on the color information, such as texture or grayscale, but only on the object geometry: whereas the first parameter could change with different luminosity, the second should be independent of it. Alignment precision defines how precisely the system overlaps the assets when it correctly tracks the object. This parameter could only be estimated by the final user on a range of 0.00 - 1.00, where 0.00 means the object is not recognized and 1.00 means that all the assets are always perfectly aligned to the real object.

The tests were performed with the applications deployed through Metaio Creator (v.3.2.2) and a Logitech HD Webcam C310 to track the printer. Table 4.1 shows the results of the tests performed for the four steps of the printer maintenance procedure, with different luminosity condition and tracking configuration. For each test, Tracking Object Quality (TOQ), Recognition Threshold (RT) and Augmentation Quality (AQ) are reported. In some cases (identified with a *), Metaio could not provide an exact evaluation of the Tracking Object Quality because the 2D image contains transparency.

Table 4.1 Tracking Tests' Results - Step 1 to 4

Tracking Configuration	LUX 40			LUX 160			LUX 300		
	TOQ	RT	AQ	TOQ	RT	AQ	TOQ	RT	AQ
Step 1									
3D CAD	3	0.70	0.80	3	0.70	0.80	3	0.70	0.80
2D Image (Render)	2*	0.30	0.00	2*	0.30	0.00	2*	0.30	0.00
2D Image (Photos)	2*	0.30	0.00	2*	0.50	0.40	2*	0.35	0.20
Step 2									
3D CAD	3	0.70	0.90	3	0.70	0.90	3	0.70	0.90
2D Image (Render)	2*	0.30	0.00	2*	0.30	0.00	2*	0.30	0.00
2D Image (Photos)	2*	0.30	0.00	2*	0.45	0.35	2*	0.30	0.00
Step 3									
3D CAD	3	0.70	0.90	3	0.70	0.90	3	0.70	0.90
2D Image (Render)	2*	0.30	0.00	2*	0.30	0.00	2*	0.30	0.00
2D Image (Photos)	2*	0.60	0.65	2*	0.85	0.90	2*	0.75	0.80
Step 4									
3D CAD	3	0.70	0.90	3	0.70	0.90	3	0.70	0.90
2D Image (Render)	2*	0.30	0.00	2*	0.30	0.00	2*	0.30	0.00
2D Image (Photos)	2*	0.60	0.65	2*	0.85	0.90	2*	0.75	0.80

The first evaluation is that the 2D image tracking based on the renders of the model does not work properly for the proposed real object. The tests were performed changing the recognition threshold from 0.70 to 0.30, to check if a feeble value could provide some significant results for the alignment quality parameter. The main issue could be that the 3D model of the printer does not provide enough texture or visual references for the tracking algorithm to match the real object. The 2D image tracking based on the photos provides better results but it is not robust enough. First of all, if the photo does not provide rich texture data (step 1 and 2), the recognition threshold drops significantly. As a consequence, also the alignment quality drops: alignment errors may occur in terms of deviation between the real object and the expected virtual asset position. Moreover, the alignment might change when moving the camera from the tracking view. This problem worsen when the luminosity is changed: in step 2 the recognition fails for darker or lighter environment conditions; in step 1 it is necessary to reduce significantly the recognition threshold to obtain a minimum alignment in the lighter environment. In step 3 and 4, the printer provides better photos in terms of recognition algorithm and the alignment is possible even when the luminosity parameter changes. Finally, this solution is still feeble if the object does not provide enough texture data and it would need an algorithm to dynamically change the recognition threshold on luminosity variation. On the other hand, this solution provides the simpler and faster way to create the tracking object, as taking the photos and doing some image editing are the only requirements. Moreover, it does not require modeling skills to produce a model of the real object, which takes more time and could be onerous. The 3D CAD tracking is the most robust solution. Changing the luminosity of the environment does not affect the alignment quality, without the need to reduce the recognition threshold. Moreover, this system provides the best alignment quality results in each step of the procedure and it is the more suitable for the maintenance domain. On the other hand, the 3D CAD-based tracking is the most onerous system because it needs a model of the real object to properly work. If the manufacturer could not provide a 3D model of the object, it is necessary to create it with a 3D modeling software: this could be difficult depending on the complexity of the object and the skills of the user. Furthermore, 3D modeling could not be possible for a variety of reasons: the original model could not be available, the object could be too tiny or simple to provide suitable recognition features and the tracking object could be too complex to provide a 3D model similar enough to the real one. To further evaluate the robustness of these solutions, other tests were performed for the three proposed luminosity conditions: the first test consisted of casting shadows over the real object in order to change the luminosity of the surface not linearly (Figure 4.1). Whereas the 3D CAD tracking was not affected by this variation, with the 2D image tracking the alignment quality would drop even more, eventually leading to not recognize the real object. The second test consisted of lightening the real object with a torch, a Maglite 3CELL D flashlight (Figure 4.2): with the 2D image tracking the result was the same as the previous test; the 3D CAD tracking was sometimes affected, especially if the torch highlighted a section of the real object that defined the geometry (such as an edge section corresponding to the line object).

Fig. 4.1 Casting shadows over the real object.

Fig. 4.2 Lightening the real object with a torch to change the surface's luminosity.

4.2 Evaluating the AMF

In order to evaluate the proposed AMF, the latter has been used to support a real case proposed by Fidia, a company that designs and manufactures sophisticated machine tools (e.g., molding machines). The Fidia's training program depends both on specific knowledge and experience of technicians to be trained and on the machine tool of interest. Usually, the first level of training, proposed to inexperienced technicians, starts with the study of available manuals. After this first phase, practical exercises are proposed both by training in laboratory and training on real case situations; exercises are performed both at the production factory and at customers' sites. The time required by this two training phases may vary considerably and the second one may last from one to three months, depending on the specific tasks requested to the technicians. The training is also different between installer technicians, who perform the initial setup of the machine, and assistance technicians, who interact with customers when problems show up and need a deeper focus on problem solving skills. In the past, there were instructors that had the duty to teach the other technicians the procedures required for each available systems. As the number and complexity of available systems increased during the years, the choice for training was to support new technicians beside skilled ones for the same task/system. Overall, the cost issues related to a specific procedure consist of three elements:

1. the time spent by the technician to learn the procedure;
2. the time spent by the technician to perform the procedure;
3. the number of errors that occurs performing the procedure, that could increase the time needed to perform the ongoing procedure or lead to further assistance requests.

These three elements also represents the performance indicators that should be used to evaluate the training system, as specified by the companies involved in the $EASE - R^3$ project.

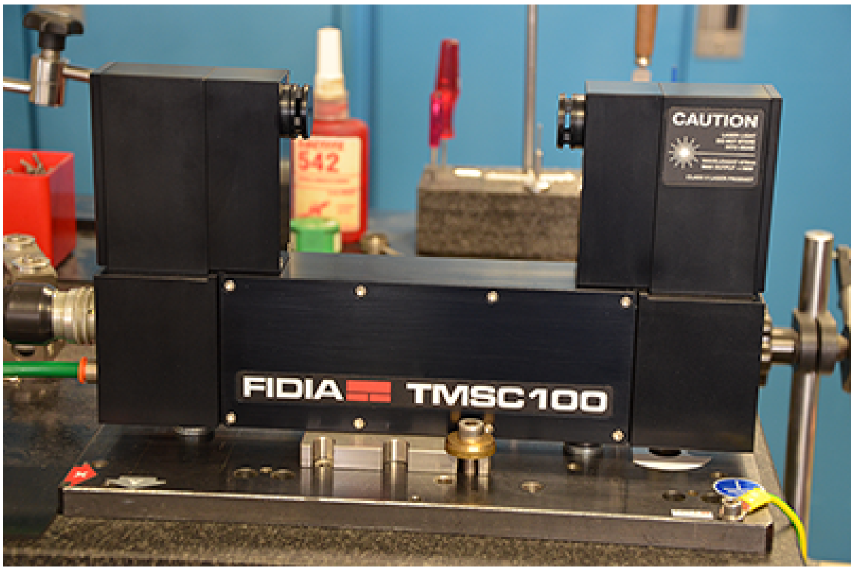

Fig. 4.3 The Fidia's TMS tool.

4.2.1 User Tests

For this case study the proposed procedure is the lenses cleaning of the Fidia's TMS (Figure 4.3). A TMS is a tool that measures, through a laser beam, the condition of molding tools in order to evaluate their precision during the lifespan and eventually suggest their replacement. Usually, these tools are used in industrial context where dust, chippings and other scraps from the machinery processing can fill or cover the laser lenses. In this situation, a specific procedure to clean the lenses of the TMS is necessary to restore its working state. To evaluate the AMF, two procedures to perform the lenses cleaning of the TMS have been proposed: the first one is a shorter and easier version of the procedure, whose aim is to evaluate if untrained, inexperienced people, who never practiced in such a field, could perform the proposed procedure in a meaningful way (Figure 4.4). The second one is a longer, more difficult procedure proposed to former technicians untrained on the specific topic; in this case, the purpose is to evaluate if the AMF could speed up the learning process necessary to train new technicians and other meaningful parameters such as its usability.

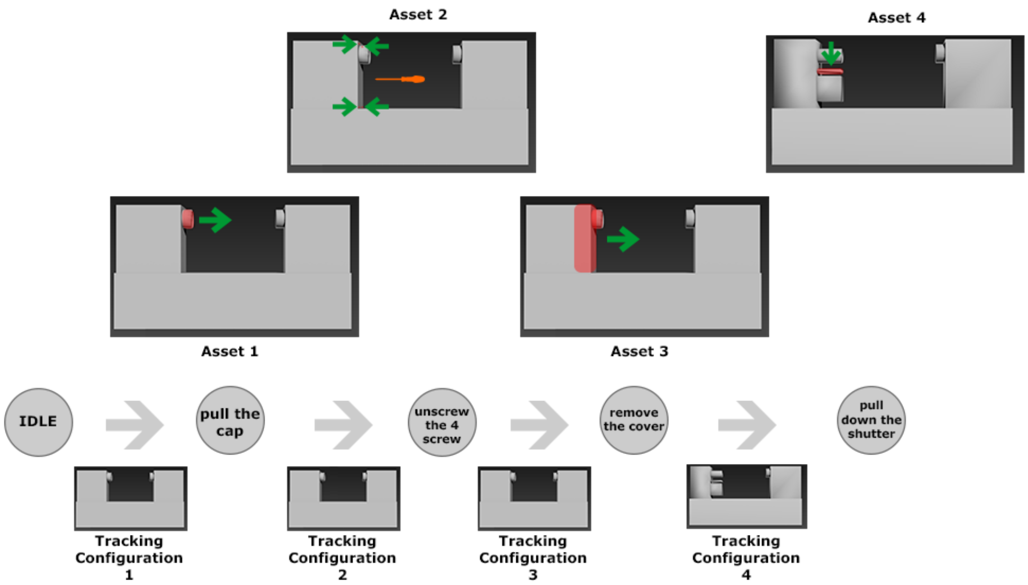

Fig. 4.4 A state machine diagram showing the short version of the procedure: for each state some assets and the tracking configuration to be recognized in order to move on to the next state are shown.

The first procedure consists of the following four steps:

1. remove the cap;
2. unscrew the four screws;
3. remove the external cover;

4. pull down the shutter to expose the lenses for cleaning.

When the lenses are reachable, in the real case, they should be cleaned using compressed air from an air can. The second procedure, which is simply an extended version of the first one, adds the following ten steps, for a total of fourteen steps:

1. clean the lenses using compressed air from an air can; pull up the shutter;
2. unscrew the shutter's crew and remove it;
3. remove the lens' cover;
4. clean the lens with a soft cloth;
5. put back the lens' cover;
6. put back the shutter in position and screw the screw that hold it;
7. put back the outer cover;
8. screw the four screws that hold in position the outer cover;
9. put back the cap.

4.2.2 Methodology

To get the users ready to the practice test, the preliminary step was to briefly illustrate them the logic of the whole system. This step required no more than 10 minutes. All the users were instructed singularly, to be sure they did not forget anything before their turn to perform the practice test. Each user performed the test alone, therefore they did not acquired any experience from viewing other participants to the test. The training to the system consisted of the following steps:

1. a brief explanation of the generic task the user should perform;
2. tools available to perform the practice are shown;
3. kind of assets the AR system provides are presented;
4. the user interface of the Android client application is presented;

5. the user interface of the Windows client application is presented (the list of the vocal commands);

6. each vocal commands is singularly presented to the user;

7. the user are assisted in wearing the AR glasses in order to maximize the comfort, the field of view and the visibility;

8. the user performs a sample tracking step, thus experiencing computer- generated assets;

9. the communication with the instructor's remote station is tested.

Users started the practice with the AR glasses device (Figure 4.5). During the practice a qualified instructor supervised the user operations without interfering, just to write down the execution time of the practice and the number of errors committed. Another instructor, placed in another room, monitored the operations through the remote station, thus waiting for help requests from the users. After completing the procedure with the AR glasses, users were also requested for repeating the procedure by using the tablet: this was necessary to evaluate advantages and drawbacks of a hand-free AR-solution, less comfortable in terms of wearing, with respect to a much handy device such a table, which instead slows down the practice when two hands are needed to perform the steps of the procedure.

Fig. 4.5 A student technician performing the training procedure on the TMS tool by the AR-glasses client.

The AMF was tested providing two groups of trainees by a pair of AR-glasses and an Android tablet with the client application, instructing them, one by one, as explained before. A first session of tests was performed with a group of 13 trainees (7 male and 6 female) enrolled in the BSc degree in Visual Design. The aim was to check the overall framework functioning, evaluate the proposed system and verify if a group of people with no experience (and a completely different background) was able to complete the procedure (short version). A second group (8 males), selected among Fidia's technicians and not previously trained for the specific task, performed the longer version of the procedure. Completion times and error rates were measured for both groups. After performing the test, trainees have been requested to compile a questionnaire to evaluate their experience.

4.2.3 Results

Table 4.2 and table 4.3 show the answers gathered by the questionnaires proposed to the two groups of trainees. Table 4.2 shows the results of a questionnaire proposed to all the participants (13) of the first test sessions; table 4.3 shows the results of a questionnaire proposed to all the participants (8) of the second test sessions. The tables present the average value of the answers (AVG), the maximum allowed value (M) and the variance (VAR); a higher value means a positive response to the question and a lower value a negative one. Although both groups of trainees have found hard (average values: group one = 3.31 and group two = 4.12) to perform the lenses cleaning task, every candidate of both groups was able to complete the assigned procedure. Even if most of the users had not previous experiences with AR applications (question 1), the overall evaluations of the framework and of the practice experience were over the average.

Figure 4.6 shows that while the first group of users composed of university students lacking of a technical background, was more prone to errors, most of the trainees of the second group performed the practice without any mistake. Evaluating the time spent by the first group of users (Figure 4.7), what stands out is that there is a wider distribution from the average value of 08 min and 38 sec., as some user had a better aptitude for the requested task or for the AMF and performed the practice very quickly; on the other hand, others did not adapt quickly to the system. In the second group of users the values of distribution are more close to the average value of 08 min and 06 sec.; in this case, the technical background of trainees smoothed over differences among trainee performances.

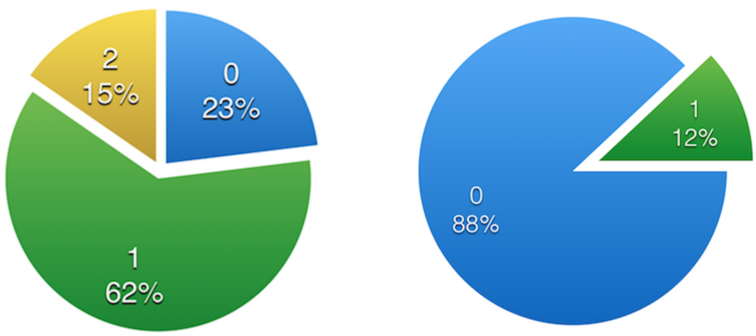

Fig. 4.6 The pie chart on the left shows the distribution of errors made by the participants of the first group, while the second one shows the distribution for the participants of the second one; the numbers over the percentage values represent the error occurrences.

Table 4.2 Questionnaire Results (1)

Questions	AV	M	VAR
1. Have you ever used augmented reality (AR) applications before?	1.15	3	0.13
2. How familiar are you with maintenance or assembly tasks (e.g. assembling IKEA furniture, repairing bicycles)?	3.31	5	1.75
3. Did you accomplish the required task?	2.23	3	0.17
4. How do you feel about the length of time required to complete the task?	3.62	5	0.24
5. How do you feel about the level of commitment needed to complete the task?	3.31	5	1.14
6. How difficult did you find the execution of the procedure?	3.31	5	0.82
7. How comfortable did you find the AR device?	3.15	5	1.51
8. How easy did you find catching the 3D model target (alignment for enabling the procedure execution)?	4.23	5	0.95
9. How did you find the alignment of the 3D model with the real object?	4.15	5	0.44
10. How effective did you find the interaction/navigation through the procedure?	3.31	5	0.98
11. How did you find the graphics of the AR device (e.g. visualising 3D elements: contrast, brightness, clearness)?	3.15	5	0.90
12. Do you think the AR device would benefit from audio/video tools supporting the procedure?	4.84	5	0.28
13. How did you find the usability of the video support tools?	4.08	5	0.84
14. How did you find the usability of audio support tools?	4.23	5	0.95
15. Do you wear glasses?	0.15 (2)	13	0.13
16. If you wear glasses, did you feel your glasses interfered with the procedure?	3.5	5	0.25
17. How tired were you after completing the procedure?	3.61	4	0.24
18. Do you think you would now be able to complete the procedure without the AR support?	4.62	5	0.85

Table 4.3 Questionnaire Results (2)

Questions	AV	M	VAR
1. Have you ever used augmented reality (AR) applications before?	1	3	0
2. How familiar are you with maintenance or assembly tasks (e.g. assembling IKEA furniture, repairing bicycles)?	4.12	5	0.61
3. Did you accomplish the required task?	2.62	3	0.23
4. How do you feel about the length of time required to complete the task?	3.87	5	1.11
5. How do you feel about the level of commitment needed to complete the task?	3.75	5	1.19
6. How difficult did you find the execution of the procedure?	4.12	5	0.61
7. How comfortable did you find the AR device?	3	5	0.5
8. How easy did you find catching the 3D model target (alignment for enabling the procedure execution)?	3.87	5	0.36
9. How did you find the alignment of the 3D model with the real object?	3.87	5	0.36
10. How effective did you find the interaction/navigation through the procedure?	4.37	5	0.98
11. How did you find the graphics of the AR device (e.g. visualising 3D elements: contrast, brightness, clearness)?	3.12	5	0.86
12. Do you think the AR device would benefit from audio/video tools supporting the procedure?	4.85	5	0.12
13. How did you find the usability of the video support tools?	3.87	5	0.86
14. How did you find the usability of audio support tools?	3.87	5	1.11
15. Do you wear glasses?	0.125 (1)	8	0.11
16. If you wear glasses, did you feel your glasses interfered with the procedure?	3	5	-
17. How tired were you after completing the procedure?	3.87	4	0.11
18. Do you think you would now be able to complete the procedure without the AR support?	4.12	5	1.11

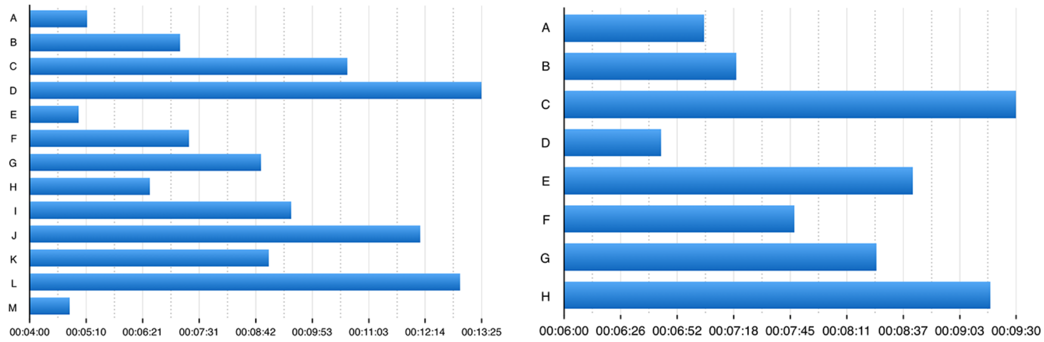

Fig. 4.7 The bar chart on the left shows the time spent by the participants of the first group performing the short procedure, while the second one shows the distribution for the participants of the second group performing the complete sequence of steps.

Moreover, all the users believed to have acquired enough experience to successfully repeat the procedure without neither the help of the AR application nor the support of an expert technician. The possibility to open a video and audio channel with the instructor operating at the remote station helps users to overcome some problems, requesting assistance to the instructor when needed and allowing the instructor to overview the procedure's fulfillment.

4.2.4 Remarks

The answers to the other questions provide useful indication about the usability of the system: higher values of variance in the evaluation of the assets and the user interface of the AMF (questions 8 to 14) point out which aspects could be improved, even if this kind of results may depend on an excessive user expectation for a new technology not experienced before. Answers to questions 16 and 17 show that the proposed AR-glasses are not the best available option for user that wear glasses on their own. As we got only 3 results in the two test sessions, this issue should be further investigated in the future. Finally, users believed that the two proposed devices, the AR glasses and the tablet, could offer the same experience in terms of effectiveness of the practice and task completing. The main point is that, considering advantages and drawbacks between hand-free operability and wearing comfort, users believed that both devices could be useful depending on the task to be performed and the operational environment. Moreover, Fidia's technicians proposed to build up a support for the tablet made of a magnetic hook and a mechanical arm to position the tablet near the focus point of the procedure, thus performing hand-free the steps of the procedure. Overall, the most important result is that each user was able to complete the assigned procedure, without previous preparation. This fact is a substantial proof of the effectiveness of the system, as it fulfills the first performance indicator. Unfortunately, it was not possible to evaluate the other two performance indicators due to the lack of data for the standard training system: neither FIDIA nor WIRES could provide information on the time spent by the technicians to perform the procedures, or statistical data on the number of errors that occurs performing the procedure during training sessions, with previously adopted methodologies. However, it is possible to identify which requirements, among the list depicted in section 2.6, have been successfully addressed by the proposed solutions, as illustrated in Figure 4.8.

REQUIREMENTS	TABLET	AR GLASSES
MUST		
Allows the user to move through various stages of the procedure.	✓	✓
Real-time support.	✓	✓
Hands-free.		✓
Off-line usage.	✓	✓
Multiple languages.	✓	✓
Menu with a list of available procedures.	✓	✓
SHOULD		
Use of glasses for augmented reality built into helmet.		✓
Don’t stop maintenance to obtain information.		✓
The application recognize the environment machine.		
View support while performing the procedure (having hands free).		✓
Video Capabilities (recording different formats).		
Physical interaction (touch).	✓	
Good Connectivity.	✓	✓
Textual assets.	✓	✓
Visual assets.	✓	✓
Not real-time support is dynamic (virtual).	✓	✓

Fig. 4.8 Requirements addressed by the proposed solutions.

4.3 Evaluating the Intuitive Interface Framework

The following section assesses the effectiveness of the proposed IIF described in 3.4.2 in supporting the selected case in terms of wearable AR-based maintenance and repair operations. Assessment encompassed three experimental tests. The first test has been designed to evaluate the IIF capability to create an interface where icons used are capable to evoke voice commands that, when issued by the user, will be recognized and will activate the expected functionality. The second test was carried out to measure the first-time user experience (FTUE) of the automatically-generated interface. In this case, intuitiveness and usability were evaluated through a set of performance indicators by comparing the first-time use of the application with a trained/experienced use. The third test was carried out to compare the FTUE of the interface created by the proposed framework with that of a suboptimal fully-personalized one. In this case, the users were allowed to freely decide the set of icons to be used for building the graphic interface, as also the mapping between voice commands and application functionalities. To ensure fairness in comparison, expansion techniques were applied to both the command sets. The hypothesis to be verified is that the first-time user performance of the interface generated with the devised framework is comparable to the trained user performance that can be achieved on the same interface, by using the sub-optimal interface.

4.3.1 Setup

For this study, 45 participants (24% female, 76% male) of an average age of 25 years (sigma = 4.8) were selected among university students and equally distributed among the three tests. Less than half (44%) of the participants had prior experience with speech recognition systems (mainly on smartphone devices and gaming consoles) and a few of them (11%) with AR applications and/or wearable devices. Considering this information, the participants were evenly distributed among the three tests. None of them had any experience in maintenance and repair of machine tools. Though all the participants were native Italian speakers and the Italian language was adopted for the speech recognizer, the rate of command recognition could still be influenced by other factors, such as the difference between the backgrounds and origins of some participants. However, these participants were evenly distributed among the three tests. As a consequence, it was assumed that these differences in speech recognition performance did not influence the overall considerations. The setup for each test is described in detail below. Three different kind of tests have been designed in order to cover all the possible situations: an introductory test to let the user familiarize with the interface; a test with an automatically generated interface; a test with an interface manually configured by the user.

First Test

This test included a preparatory phase, in which the users were driven by an instructor into exploring the automatically-generated icon-based interface. The icons were indicated to the user, in a predefined order, with the aim of activating all the functionalities in the state machine. The user was asked to activate the functionality associated to each icon by choosing commands that he or she feels are evoked by the icon. In the following, this phase will be referred to as $T1_{autom}^{(prep)}$, where superscript $(prep)$ refers to the preparatory nature, and subscript $(autom)$ to automatic creation of the interface.

The core of the first test was actually represented by the second phase in which, after the above training, the user was asked to complete the whole maintenance procedure without any direct supervision. In particular, the task consisted in performing regular maintenance of a high-precision laser meter. The task requires the user to follow a sequence of steps, from unscrewing and removing some parts of the tool to cleaning the internal lens. The user was shown a video that visually explains the operation [1]. Since the users already got acquainted with the interface during the first phase of the test, they could be considered as experienced users for the second phase. Hence, in the test that follows, this phase is referred to as $T1_{autom}^{exp}$, where exp stands for experienced users.

Second Test

In this test, the users were verbally introduced to all the available functionalities, but they were not apprised about the available icons or about how to activate the associated functionalities (i.e., which are the commands recognized). After such introduction, they were asked to wear the AR glasses and to autonomously complete the above mentioned maintenance procedure by looking at the icons in the automatically-generated interface, deducing the associated functionalities, activating them and performing the operations suggested by the AR hints. Because there was

[1] https://youtu.be/xAL0w4LgiIo

no training on the interface, as in the first test, this test is referred to as $T2^{ft}_{autom}$, where ft refers to first-time users.

Third Test

In the preliminary step of this test, the users were asked to manually create their own preferred interface. To this end, the users were presented with a description of all the application functionalities, and then asked to select, for each functionality, an evocative icon and a list of voice commands. The commands were lexically and semantically expanded. Afterwards, the users were asked to complete the same maintenance procedure as that of the previous tests by using the customized interface in an autonomous way. Hence, this test is referred to as $T3^{ft}_{custom}$. The ft superscript indicates that, as in the previous test, first-time users are considered, whereas the subscript $custom$ refers to the use of the customized interface.

4.3.2 Assessment Strategy

For each test, both objective and subjective measures were collected during interaction with the AR application, as well as after the execution of the maintenance procedure. During the test, a report on system behavior was produced for each participant by logging the data about system behavior and recording the associated audio tracks containing the commands issued. A cross-analysis between data logs and audio tracks was carried out to extract quantitative information about user interaction and performance of the speech recognition module, including false positive and false negative error rates. Additional objective measurements made during the test include the average number of total attempts for activating a given functionality and the task completion rate. For the supervised phase in Test 1, a task was considered completed if all the functionalities corresponding to the icons indicated to the user were correctly activated. For this purpose, the maximum number of attempts per functionality was defined. When the system was used autonomously to carry out the designed task (i.e., in the second phase of Test 1, and in Tests 2 and 3), the task was considered completed if the participant succeeded in reaching the last step of the procedure without any help from the supervisor, say when he or she failed to activate a critical functionality, using no more than the finite number of attempts. A functionality is considered critical when, if not activated, it would not be possible to complete the task; for the given use case, critical functionalities are “procedures”, “confirm”, “start procedure”, “next step” and “exit”.

After the test, each participant was asked to evaluate the usability of the speech interface through a subjective questionnaire, based on SASSI methodology [233]. The questionnaire had 34 statements to be evaluated on a 7-point Likert scale. The statements refer to six usability factors, as depicted in Figure 4.9: System Response Accuracy (SRA), Likeability (LIKE), Cognitive Demand (CD), Annoyance (AN), Habitability (HAB) and Speed (SPE).

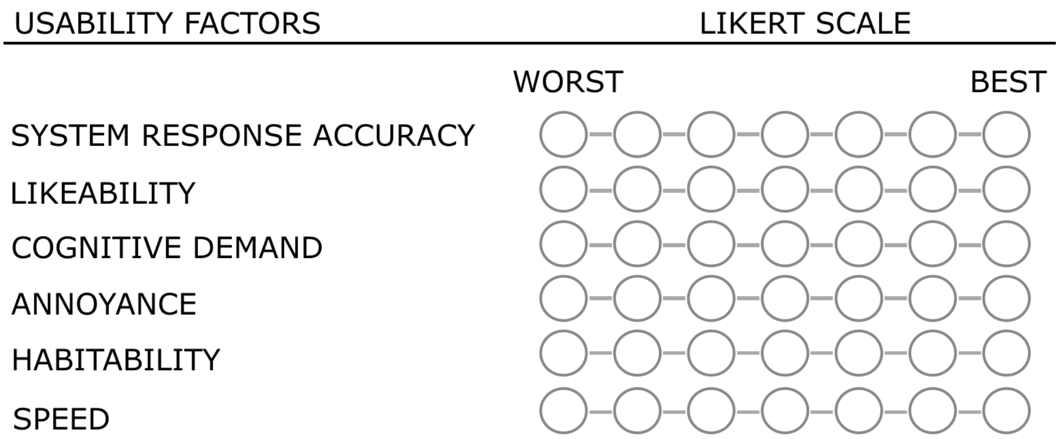

Fig. 4.9 SASSI usability factors.

4.3.3 Evaluation Metrics

The tests were aimed at evaluating the number of recognized functionalities. A functionality is considered recognized whenever a participant succeeds in activating it. However, during the activation, several possible situations may occur, such as true positives (TP), true negatives (TN), false positives (FP) and false negatives (FN). In the following paragraphs, each situation is defined considering the perspectives of both the user and the system, which, in some cases, may differ. The user perspective considers the expected system behavior when a command is issued, whereas the system perspective considers the behavior of speech recognition.

A true positive (TP) occurs when a functionality is activated and, in fact, this is the one the user wanted to activate. This indicated that the system worked properly and the user perceived it as the correct behavior. This is the case, for instance, of a user who wanted to disable the vocal user interface. He or she saw an icon with a banned microphone and said “spegni il microfono” (“turn off the microphone”). The utterance was correctly recognized (correct behavior of the system) and mapped on the corresponding command, which activated the right functionality.

False detections (both positive and negative) can be divided into two categories based on the causes of their occurrence. When, because of a failure in the command expansion, the system activates the wrong functionality, the false detection is attributed to a semantic cause (SC). False detection can also be attributed to a technological cause (TC) when, for instance, noise or possible configuration issues negatively affect the performance of the speech recognizer. A system error that could not be attributed to a semantic cause is then attributed to a technological cause.

A false positive (FP), or command switch, occurs when the activated functionality is not what the user expected. This kind of error occurs in two situations. The user might have pronounced the command that is actually mapped to the intended functionality but, for some reason, the system recognized a different command, mapped to another functionality (FP_t). This may happen because of ambient noise or poor performance of the voice recognition module. Command switches also occur when the user pronounces a command that is not associated to the intended functionality (FP_s). For instance, to disable the voice interface, the user might say “volume” (“volume”), to which the system would respond by changing the headset volume.

A false negative (FN) occurs when the user tries to activate a functionality in the correct way (i.e., a command that should have been recognized the way it was issued), but the system fails to recognize it and, at the same time, the command it recognized, if any, has not been routed to activate another existing functionality. False negatives can be due to a technological reason (FN_t), e.g., when recognition confidence is too low. They could also occur when the speech recognizer returns a result that does not correspond to the user’s command. For instance, the user might say “video” (“video”), but because of assonance, the system recognizes the command as “vedova” (“widow”), which is not mapped to any functionality. On the other hand, the reason for false negatives could be semantic (FN_s) when the utterance is not associated to any valid command. For instance, the user might want to start reproducing a video by saying “riproduci” (“reproduce”), but command to be used is “play video”.

Finally, a true negative (TN) occurs when the system does not map the command issued to any functionality. The recognition is not successful, because the pronounced command is not valid. The icon is probably not appropriate, although the system worked properly. This could be the case, for instance, when the user sees an icon with a plus sign, which is actually mapped to the functionality to increase the headset volume, but the user says "zoom" ("zoom"). However, from the user's point of view, a true negative cannot exist, since every time he or she issues a command associated to a given icon, he or she expects the system to activate the corresponding functionality. Therefore, in the user's perspective, a true negative is considered a false negative with a semantic cause (FN_s).

4.3.4 Objective Evaluation

The objective data gathered during the tests is summarized in two separate tables to simplify understanding and comparison. Table 4.4 presents the results of the preparatory phase of Test 1 ($T1_{autom}^{(prep)}$), which can be analyzed to assess system's ability to create an interface with evocative icons by observing user's operations in supervised conditions. The data in this table cannot be compared with the data pertaining to the second phase of Test 1 or with the data collected in Test 2 and Test 3, where the users were asked to interact with the system in an autonomous way. The data relating to autonomous interaction is presented in Table 4.5. In this case, the data describes the execution of the same maintenance procedure under three different conditions, i.e., by experienced ($T1_{autom}^{exp}$) or first-time users ($T2_{autom}^{ft}$ and $T3_{custom}^{ft}$), using the automatically-generated (subscript $autom$) or by the customized (subscript $custom$) interface. In both cases, the performance of automatically-generated interface is measured as the ability of the system to select icons that can evoke proper mapping with recognized commands and expected functionalities, expressed in terms of correct behavior (TP), errors (FP and FN), completion rate (CR) and average number of attempts per functionality (n_a).

Table 4.4 Tests' Results (1)

Test	TP	FP_s	FP_t	FN_s	FN_t	CR	n_a
$T1_{autom}^{(prep)}$	69,1%	2,3%	0,6%	14,8%	13,1%	73%	1.50

Table 4.5 Tests' Results (2)

Test	TP	FP_s	FP_t	FN_s	FN_t	CR	n_a
$T1^{exp}_{autom}$	83,8%	1,5%	1,0%	2,5%	11,1%	86%	1.18
$T2^{ft}_{autom}$	71,9%	0,9%	0,3%	15,7%	11,1%	85%	1.42
$T3^{ft}_{autom}$	63,4%	2,5%	0,4%	30,6%	3,1%	60%	1.61

In executing the tests, each interaction was manually annotated to keep track of the errors and correct behaviors, which were classified, based on the user's perspective as defined in section 4.3.3. For each test, columns 2–6, in both the tables, give the number of true positives (TP), semantic and technological false positives (FP_s and FP_t), and false negatives (FN_s and FN_t), whereas columns 7 and 8 present the average task completion rate CR and the average number of attempts n_a required for activating each functionality.

During the preparatory phase of Test 1 (Table 4.4), the task completion rate was 73%, indicating that, on average, three users out of four were successful in activating all the functionalities in the automatically-generated interface. The participants committed errors mainly during the last phase of the task, and the errors belong to the false negative category, i.e., the participants were not successful in immediately activating a given functionality, consequent to which their commands were ignored by the system. This kind of error was equally distributed between the semantic (FN_s) and technological (FN_t) false negatives. Semantic false negatives occurred, for instance, when the user completed a particular step of the maintenance procedure and wanted to proceed further by activating the "next step" functionality, but said "fatto" ("done"), but the command was ignored by the system. Technological false negative occurred, for instance, when the user tried to open the settings panel by using the command "proprietà" ("properties"), which was falsely recognized by the system as "attività" ("activities").

In this case, false positives were scarce (2.9% of the total fifteen attempts) and this holds good for other scenarios too. In general, 2.3% of the command switches were FP_s errors. These errors occurred mainly in two practical situations: (i) When some participants used the command "opzioni" ("options") to activate the functionality for displaying the list of available procedures, and the system reacted by opening the settings menu, rather than the expected list; (ii) When the participants used the command “parti”, in Italian, to activate the maintenance procedure, which means either “start” or “leave”, depending on the context to which the system responded by erroneously activating the functionality for exiting from the current state. Technological errors FP_t constituted only the 0.6% of the total number of attempts. An example of misunderstanding by the speech recognizer is distortion of a command like “ripeti” (“repeat”) into an existing command like “parti” (“leave”) and again forcing the exit from the current state.

The intuitiveness of the proposed user interface can be evaluated by comparing the results obtained by first-time users operating with the automatically-generated interface in $T2^{ft}_{autom}$ with those obtained by the two control groups represented by the users involved in $T1^{exp}_{autom}$ and $T3^{ft}_{custom}$.

In $T2^{ft}_{autom}$, the participants were asked to execute the maintenance procedure without prior training. The participants, belonging to the control group $T1^{exp}_{autom}$, were considered experienced users because of their learning in $T1^{(prep)}_{autom}$. This was confirmed by the fact that, as expected, the number of true positives largely increased between the first ($T1^{(prep)}_{autom}$) and the second ($T1^{exp}_{autom}$) phases of the first test. This trend was confirmed by the number of semantic false negatives (FN_s), which decreased drastically in the second phase of the test indicating that the number of failures in functionality activation attempts was low. First-time users of the automatically-generated interface achieved performance comparable to that of trained users. This is confirmed by the average completion rate, which is almost the same in $T2^{ft}_{autom}$ and $T1^{exp}_{autom}$ (85% vs. 86%), attained at the cost of a higher average number of attempts per functionality due to a higher number of semantic false positives (1.42 vs. 1.18, on average). The participants had to try several commands to activate a given functionality but, in the end, they completed the maintenance procedure just as experienced users.

During the test $T3^{ft}_{custom}$, the users built their own personalized interface by selecting icons and commands for each functionality. Even then, the completion rate was higher when the automatically-generated interface was used in $T2^{ft}_{autom}$ (85% vs. 60%). This is reflected by the lower number of user attempts for activating a functionality (1.42 vs. 1.61, on average), which means that, on average, the users had to issue fewer commands to complete the task with the automatically-generated interfaces than with the personalized one. Similar inferences can be drawn by looking at error percentages. The most common problems in $T3^{ft}_{custom}$ were due to the fact that the participants forgot the commands and tried to use the commands associated to other functionalities, disregarding the importance of visual cues.

4.3.5 Subjective Evaluation

Subjective usability scores collected with the SASSI questionnaire [233] were mapped on a better-to-worse scale (7-to-1) and averaged per factor. The results obtained show that the user experience was not the same during the three tests (see Figure 4.10). Considering the aggregated scores (bars labeled with TOT.AVG. in the plot), two considerations could be made intuitively.

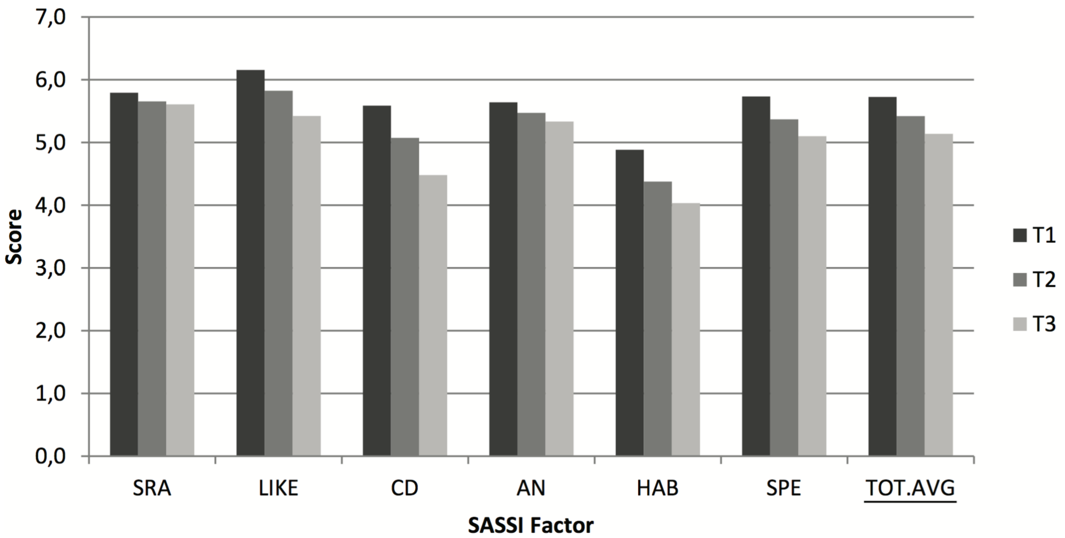

Fig. 4.10 Results of subjective evaluation based on SASSI methodology.

First, the system apparently performed better in Test 1 than in Test 2. This could be because, during Test 1, the participants had the chance to get acquainted with the system before using it autonomously. Second, in Test 3 (i.e., with the control group that used the personalized interface), the usability was lower than that in Tests 1 and 2 (i.e., when the automatically-generated interface was used). In fact, although participants were allowed to choose their preferred icons and commands, their satisfaction was low. The system failed to adequately expand the commands, and the manually-selected icons could not evoke the selected command. Consequently, only a subset of the commands issued were actually recognized by the system.

To confirm the statistical significance of the foregoing results, a one-way repeated measures ANOVA (Analysis of Variance) test was carried out. The null-hypothesis $H_0 : \mu_{T1} = \mu_{T2} = \mu_{T3}$ was that the users had the same overall experience during the three tests. The mean square between evaluations is $MS_B = 1.298$ and that within evaluations $MS_W = 0.221$. Therefore, the F statistic is $F = MS_B/MS_W = 5.88$. The results reveal that the average scores are significant at $\alpha < 0.05$ level, $F(2,42) = 5.14$, $p = 0.01$. Since the p-value obtained from the F-distribution is lower than the significance level $\alpha = 0.05$, the null hypothesis was rejected, concluding that the three tests did not have the same mean preference. A post-hoc analysis was performed by applying the Tukey HSD (Honestly Significant Difference) test to determine which of the three means are statistically different. The results of the critical values for the Studentized range statistic $Q, q(0.05,3,45) = 3.43$, reveal that only the difference between the results of Tests 1 and 3 are significant in the 95% confidence interval.

In summary, it can be concluded that the average subjective evaluations of Tests 1 and 3 are unequal, and this confirms the second intuitive observation. A preference for Test 1 with regard to Test 2, or for Test 2 with regard to Test 3, could not be found from the analysis of variance, and this confirms the hypothesis that first-time users of the automatically-generated interface (Test 2) had a user experience, comparable to the experience of both the experienced participants (Test 1) and the subjects using the personalized interface (Test 3), which is considered a suboptimal case.

In Table 4.6, disaggregated results are presented to enable a more in-depth analysis of individual factors and related statements. For each test, the mean and the variance of SASSI scores were calculated. Considering System Response Accuracy (answers 1 to 9), it can be observed that the participants perceived the automatically-generated interface (Tests 1 and 2) as slightly more accurate and reliable than the personalized interface. Likeability (answers 10 to 18) of the personalized interface was affected by the difficulty involved in remembering the chosen commands, thus rendering the automatically-generated interface to be more appealing. The users found that the automatically-generated interface is more effective for recovering from errors, besides being easier to interact with. This is confirmed by Cognitive Demand (answers 19 to 23) and, in particular, by the 23rd statement that the automatically-generated interface is easier to use than the personalized one. Concerning Annoyance (answers 24 to 28), the users perceived the automatically-generated interface as more flexible than the personalized one. As regards Habitability (answers 29 to 32), the participants felt uncomfortable with the personalized interface, because they often wondered if they were using the right words. Finally, as regards speed (answers 33 and 34), the users perceived the automatically-generated interface as faster than the personalized one.

Table 4.6 Subjective Results

	Test 1		Test 2		Test 3	
Statement	s_{mean}	σ^2	s_{mean}	σ^2	s_{mean}	σ^2
System Response Accuracy (SAR)						
1. The system is accurate.	5.60	0.97	5.62	0.42	5.27	0.92
2. The system is unreliable.	6.47	0.70	6.58	0.27	6.33	0.67
3. The interaction with the system is unpredictable.	5.87	0.84	6.17	0.52	5.87	0.84
4. The system didn't always do what I wanted.	5.33	2.81	4.58	1.72	5.67	2.52
5. The system didn't always do what I expected.	5.20	3.46	5.25	2.02	5.80	1.74
6. The system is dependable.	5.87	0.98	5,92	0.63	5.40	0.40
7. The system makes few errors.	6.07	0.92	5.42	2.27	4.93	2.78
8. The interaction with the system is consistent.	6.13	1.12	5.67	1.15	5.40	0.40
9. The interaction with the system is efficient.	5.60	1.97	5.69	1.06	5.80	0.60
	5.79	1.58	5.65	1.12	5.61	1.29
Likeability (LIKE)						
10. The system is useful.	6.33	0.38	6.42	2.08	6.13	1.12
11. The system is pleasant.	5.93	0.92	5.50	3.36	5.20	1.46
12. The system is friendly.	5.87	1.12	5.64	1.45	5.27	0.64
13. I was able to recover easily from errors.	6.13	0.84	5.67	1.52	4.60	1.69
14. I enjoyed using the system.	6.27	0.35	6.08	2.08	6.00	0.86
15. It is clear how to speak to the system.	6.33	0.38	5.33	1.15	5.33	0.52

Table 4.6 Subjective Results

	Test 1		Test 2		Test 3	
Statement	s_{mean}	σ^2	s_{mean}	σ^2	s_{mean}	σ^2
16. It is easy to learn to use the system.	6.73	0.35	6.00	1.09	5.27	1.64
17. I would use this system.	5.80	1.17	6.25	2.20	5.60	2.11
18. I felt in control of the interaction with the system.	6.00	1.14	6.00	0.91	5.40	1.97
	6.16	0.77	5.82	1.67	5.42	1.44
Cognitive Demand (CD)						
19. I felt confident using the system.	6.13	1.27	5.42	1.72	4.87	1.41
20. I felt tense using the system.	5.53	2.12	5.00	2.36	4.33	2.10
21. I felt calm using the system.	5.47	2.84	5.50	2.09	4.53	1.70
22. A high level of concentration is required when using the system.	4.60	2.40	3.83	2.33	3.87	0.70
23. The system is easy to use.	6.20	1.03	5.62	0.76	4.80	0.74
5.59	2.16	5.07	1.82	4.48	1.39	
Annoyance (AN)						
24. The interaction with the system is repetitive.	4.27	2.64	4.38	3.59	4.53	8̇4
25. The interaction with the system is boring.	5.67	2.10	5.25	2.93	5.53	1.41
26. The interaction with the system is irritating.	6.00	1.57	6.17	0.88	5.87	1.41
27. The interaction with the system is frustrating.	6.83	0.81	6.25	0.93	5.80	2.03
28. The system is too inflexible.	5.93	2.07	5.00	2.36	4.93	1.78
	5.64	2.26	5.47	1.98	5.33	1.87
Habitability (HAB)						

Table 4.6 Subjective Results

	Test 1		Test 2		Test 3	
Statement	s_{mean}	σ^2	s_{mean}	σ^2	s_{mean}	σ^2
29. I sometimes wondered if I was using the right word.	3.47	3.27	3.08	2.24	1.87	1.55
30. I always knew what to say to the system.	4.47	2.70	3.75	0.93	2.93	2.21
31. I was not always sure what the system was doing.	5.67	2.10	5.17	1.97	5.27	2.21
32. It is easy to lose track of where you are in an interaction with the system.	5.93	1.64	5.67	2.24	6.07	0.50
	4.88	3.29	4.38	2.57	4.03	4.47
Speed (SPE)						
33. The interaction with the system is fast.	5.73	1.21	5.15	1.64	4.80	1.46
34. The system responds too slowly.	5.73	1.13	5.58	1.54	5.40	1.69
	5.73	1.17	5.37	1.31	5.10	1.61

4.4 Evaluating the Speech Interface Robustness

This section presents and discusses the obtained results both from an objective and subjective point of view. Eleven people tested the four interfaces: MW, AVOS, IVOS and BVOS. The first two scanning algorithms are triggered by a single word (the equivalent of a physical switch), whereas the last scanning algorithm is based on a three-words solution, where two words are used to move forward and backward into a list of commands organized as a circular buffer and the third word is used to confirm a command selection. These are the words used in three scanning algorithms: for AVOS the Italian command is “conferma”, equivalent to the English word ‘confirm’; for IVOS the Italian command is “avanti”, equivalent to the English word ‘next’; for BVOS, the Italian commands are “avanti”, “indietro” and “conferma”, respectively equivalent to the English words ‘next’, ‘previous’ and ‘confirm’.

4.4.1 User Tests

The test requires the users to interact with an AR application for maintenance operations. The users have to navigate throughout the menus of the application and activate icons in order to try all the available functionalities. A sequence of slides displayed on a monitor instructs the users on which icon they should activate, step by step. Each user had to repeat the test four times in order to try out all the four different solutions. The users that participated to the test were both males (9) and females (2) students of the MSc in computer science at the Politecnico di Torino. Their age ranged between twentyfive and thirtythree. The users mostly declared to possess an average competence in the use of speak recognition interfaces.

4.4.2 Methodology

The eleven testers were trained individually; in particular, they were asked for performing a hands-free maintenance task. The number of steps has been previously defined and was kept equal for all tests. As testers were not professional technicians, but students of the MSc in computer science, a table of vocal commands related to each icon of the interface was provided; in this way, it has been avoided to artificially increase the mental load when tests with the MW interface were performed. Each tester tried all four solutions and then filled a questionnaire. The same interface was first tested in a quiet environment and then the test was repeated by adding an artificial environmental noise. The artificial noise was aimed to simulate the background noise in an industrial plant; the average intensity of the noise was approximately 67dB, with a maximum recorded value of 74dB. For each test, the noise track was played from the beginning in order to provide the same conditions.

Some objective parameters were recorded: the number of false positives, the number of false negatives, the number of words pronounced and the time necessary to complete the task. A false positive is considered when a wrong command is triggered; this can happen when the recognition engine confuses an environmental "sound" as a valid command or when the scanning algorithm leads to select a wrong command (this is possible, for instance, for scanning algorithms based on temporized selection mechanisms). When a right command is not recognized (for instance, when a loud background noise temporarily overlaps) the number of false negatives increases.

4.4.3 Objective Results

Number of errors, completion times and number of pronounced words are listed in Figure 4.11, 4.12 and 4.13, respectively. It is possible to notice how the performance of all the interfaces drops with the environmental noise; on the other hand, the three VOS interfaces limit the number of errors to a maximum of three. Also, a statistical analysis confirm this claim.

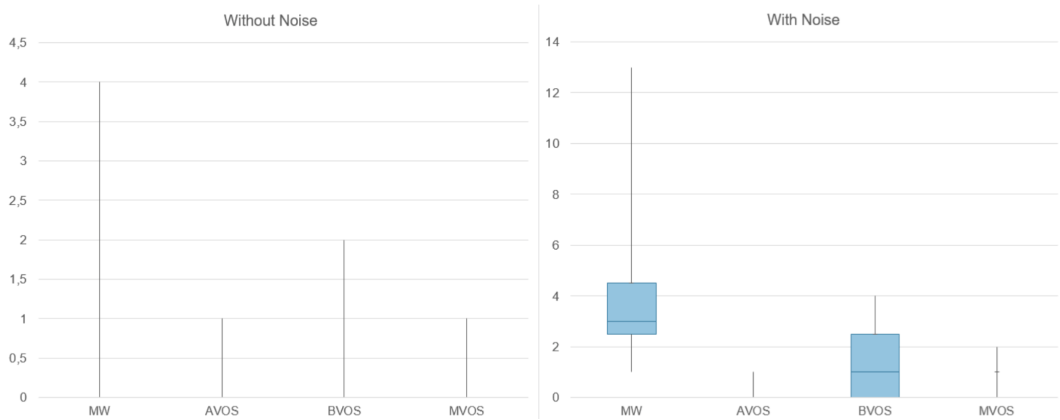

Fig. 4.11 Number of errors (false negative + false positive) with and without environmental noise.

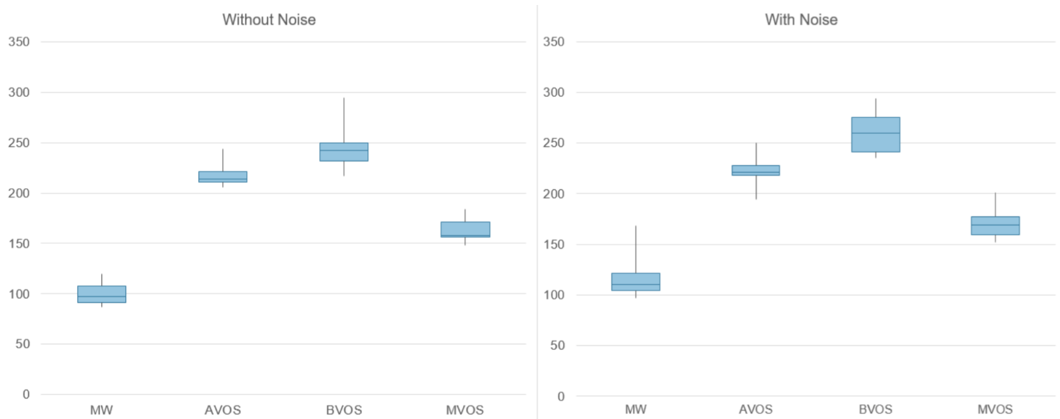

Fig. 4.12 Times (in seconds) to complete the assigned task.

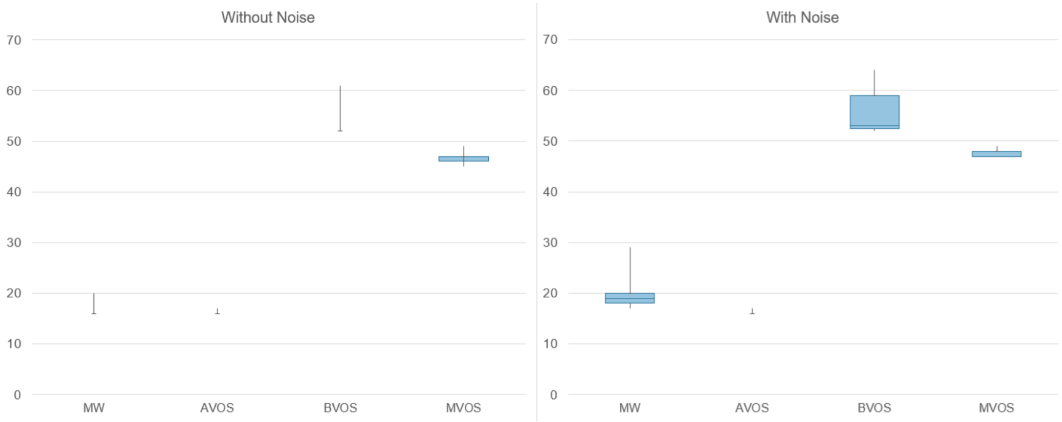

Fig. 4.13 Words pronounced to complete the assigned task.

As variances are unknown, paired t-tests are used to test the null hypothesis that the mean difference between number of errors of the MW interface and each VOS interface is equal to zero (e.g., $\mu_t = \mu_{MW} - \mu_{AVOS/IVOS/BVOS} = 0$). On the other hand, the alternative hypothesis is that the robustness of VOS interfaces is better. A level of significance has been considered: $\alpha = 0.05$. T-statistic values show how the null hypothesis can be always rejected: when MW and AVOS are compared, statistic t is equal to 3,89 with respect to a t critical equals to 2,22, whereas statistic t is equal to 2,26 with respect to a t critical of 2,13 when MW and IVOS are compared and statistic t equals to 2,90 with respect to a critical value of 2,20 when MW is compared with BVOS. The same analysis about completion times and number of words pronounced outlines how the MW interface is always better than the scanning algorithm-based solutions considered; the only exception is the AVOS interface when the number of pronounced words is considered.

If the robustness is improved by VOS interfaces, the latency time can be an issue. Automatic and inverse scanning algorithms lead to an average completion time about the double of the MW solution. On the other hand, the bidirectional scanning algorithm provides an increased robustness and a limited overhead in latency times. The BVOS interface is affected by a number of words to be pronounced that is about the triple of the MW interface. From an objective point of view, it is not easy to definitely select the best approach as a lot of other parameters should be also considered. Robustness and latency are just two dimensions of a domain where mental load, user preferences, environmental conditions and the application itself play a non marginal role.

As mentioned before, the list of commands has been provided, but this might strongly reduce the mental load really necessary to use the application. Moreover, the proposed application can be controlled by a very limited number of commands (less than 20); very different performance could be detected for more complex applications presenting several tens commands: the robustness of speech recognition-based applications generally decreases with the number of words. VOS interfaces are not affected in term of robustness as the dictionary size is constant. On the other hand, a larger number of commands entails a larger scanning (latency) time and this issue should be tackled by considering more sophisticated scanning algorithms such as the ones introduced for text entry [221].

4.4.4 Subjective Results

After completing the test, the testers were asked to evaluate their experience in three different ways. Firstly, they had to rate the interfaces in a scale between 0 (bad) and 4 (good) for five different qualities to evaluate the usability of the interfaces, as defined in [234]. The five qualities were defined as follows:

1. Learnability. "How easy is it for you to follow the proposed instructions the first time you encounter the interface?"
2. Efficiency/Effectiveness. "Once you have learned what to do, how quickly can you perform the proposed instructions?"
3. Memorability. "When returning to the application after a period without using it, how easily would you reestablish proficiency?"
4. Errors. "How many errors did you make, how severe were these errors and how easily did you recover from them?"
5. Satisfaction. "How pleasant is it to use the interface?"

The overall results of the Nielsen usability questionnaire are listed in Figure 4.14. The BVOS interface has the higher rating for learnability, memorability and errors, and it is only second to the MW interface for efficiency/effectiveness and satisfaction. Overall, it is the interface with the highest acceptability. The MW interface got the best evaluation for both efficiency and satisfaction, but it got the worst results for learnability and memorability, achieving the second place among the four available interfaces. The AVOS and IVOS are considered better than the MW for learnability and memorability but they are otherwise considered the worst solutions, with a slight overall difference between them.

Secondly, the testers had to rate the interfaces, on a scale between 0 (bad) and 4 (good), for six different qualities derived from the Subjective Assessment of Speech-System Interface Usability (SASSI) principles: “system response accuracy”, “likeability”, “cognitive demand”, “annoyance”, “habitability” and “speed” (terms as defined in [233]). These six qualities were described to the users as follows:

1. “system response accuracy” refers to the robustness of the system in recognizing the user’s input correctly and whether the system does what the user expects;
2. “likeability” means that the users enjoy using the system, perceive the system as friendly and would use it again;
3. “cognitive demand” refers to how much difficult and stressful the system is to be used and how much effort and concentration it requires;
4. “annoyance” is related to how much the system is irritating/repetitive/boring;
5. “habitability” defines the user’s confidence in what the system is doing and how to interact with it;
6. “speed” simply refers to the speed of the system.

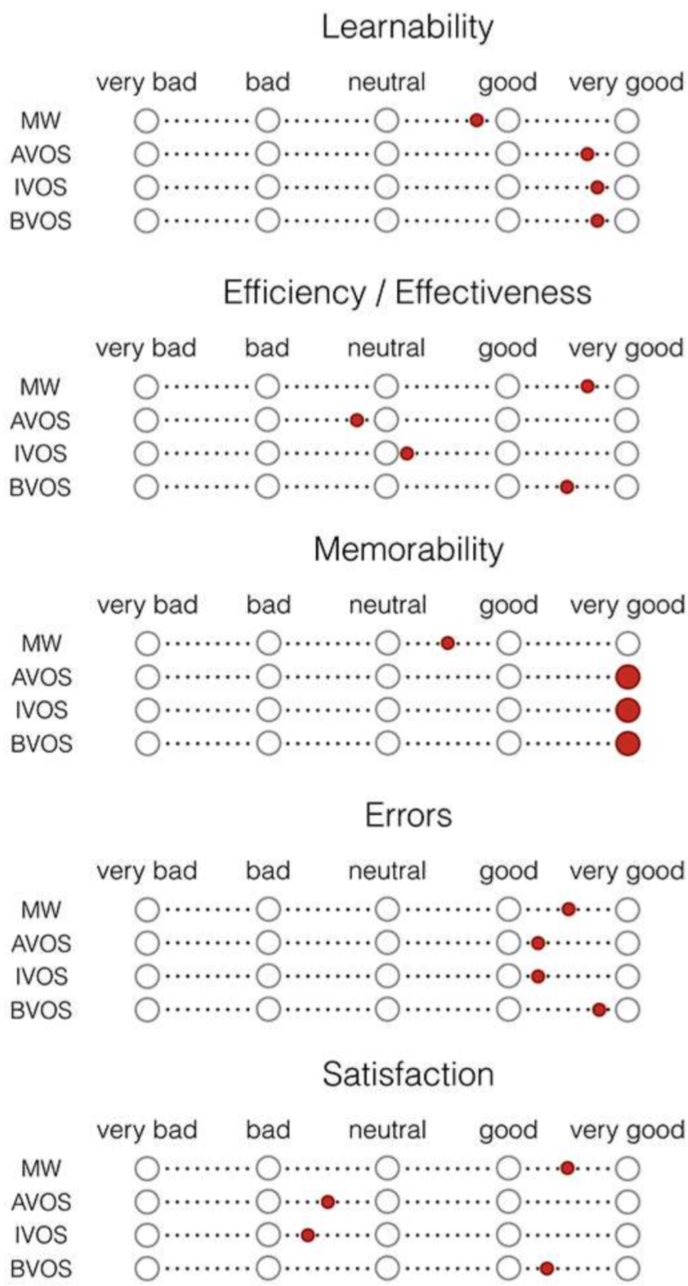

Fig. 4.14 Usability evaluation based on the NIELSEN usability principles.

The overall results of the second usability questionnaire are listed in Figure 4.15. The MW interface is perceived as the overall best solution, even if by a slight margin, and it is the better one in terms of likeability, speed and minimum annoyance. It is considered worse than the BVOS in terms of accuracy and habitability, thus the BVOS is perceived as the second best option among the four available interfaces.

The AVOS and IVOS interfaces are considered better than the other two only in terms of cognitive demands, thus they are classified by the users as the worst possible solutions, with a minor difference in the overall evaluation between the two.

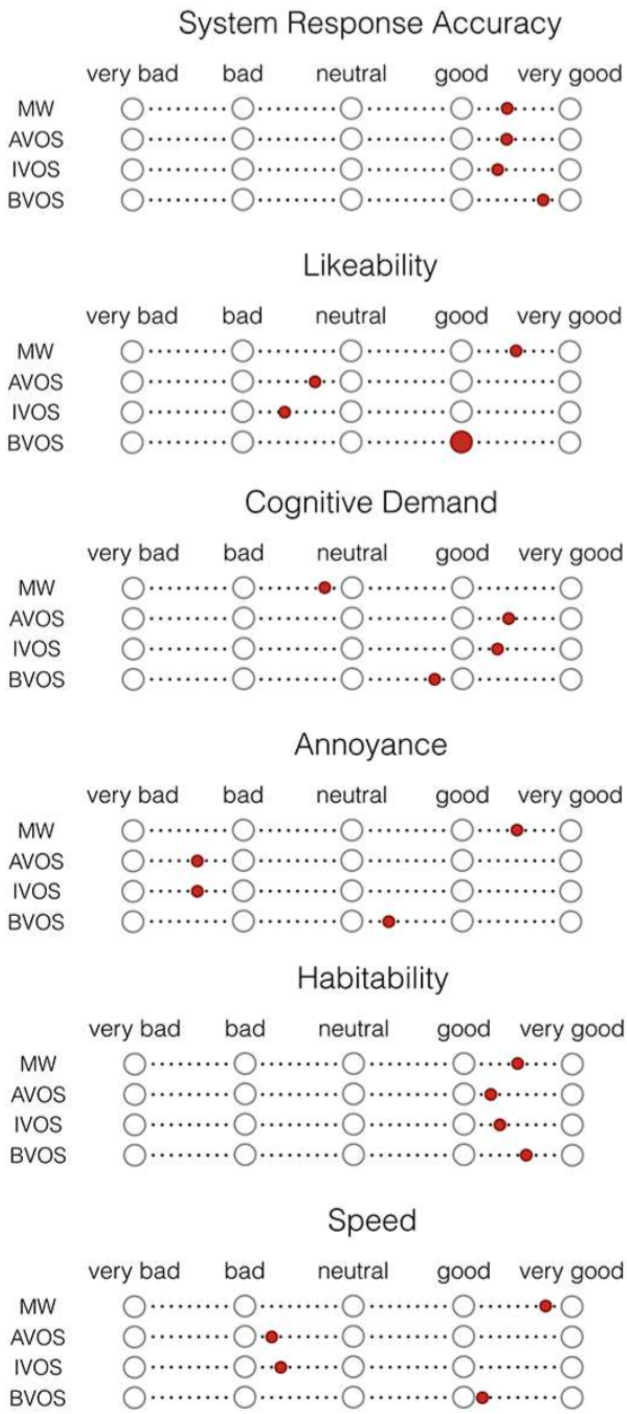

Fig. 4.15 Usability evaluation based on the SASSI usability principles.

Finally, the users where asked to rank the four proposed interfaces and to provide a motivation for their choices. Table 4.7 shows the ranking of the four interfaces, expressed as number of testers that choose each option. Seven testers out of eleven selected the MW interface as their first choice, because it is the fastest and easiest interface available, with the lowest latency value and the lowest number of commands to pronounce. Four testers selected the BVOS interface as their first choice despite of the high number of commands they have to utter, because they perceived the need to look for the correspondence between the icons and the vocal command or to learn it as a limitation. Three users depicted the BVOS system as the best alternate solution to the MW interface in terms of better reliability and lesser cognitive demand. Only one tester selected the AVOS as a valid alternative to the MW interface due to its simplicity. The AVOS and IVOS solutions were generally depicted as the worst interfaces in view of their high latency time. Two users preferred the AVOS considering its easiness. Eight users preferred the IVOS because they had more control on the interaction with the interface.

Table 4.7 Interfaces' Ranking

Interface	First Choice	Second Choice	Third Choice	Fourth Choice
MW	7	4	0	0
AVOS	0	1	2	8
IVOS	0	0	8	3
BVOS	4	6	1	0

4.5 Conclusions

This chapter describes the tests that were designed and performed to evaluate all the solutions investigated in the previous chapter. Each section describes the tests, the criteria necessary to correctly evaluate them, the methodology adopted to retrieve all the data and the evaluation of the tests results. Figure 4.16 specifies which of the remaining problems are related to the used tools and which are related to the architecture (thus providing possible future research areas).

CHALLENGES	OPEN PROBLEMS	CAUSES
POSE TRACKING	CAD Tracking of simple object Recognition of current step completion	Architectural limitation
RECONFIGURABILITY		
CONTENT CREATION	Multiple choices from a given node	Architectural limitation Tracking algorithm limitation
INTERFACE DESIGN	Recognition errors could be reduced	Architectural limitation Technological limitation
INTERFACE ROBUSTNESS	Reduce perceived annoyance Reduce errors	Technological limitation
TECHNOLOGY ACCEPTANCE	Improve usability of provided solutions	Technological limitation Architectural limitation

Fig. 4.16 Remaining problems.

In the next chapter, conclusions and possible future works are depicted.

Chapter 5

Conclusions and Future Work

Part of the work described in this chapter has also been previously published in [3] and [4], [60], [61] and [62]

Thanks to the reduced cost of technology and the increasing familiarity of average users with interactive computer and mobile applications, AR-based solutions for maintenance, repair and assembly are emerging as an interesting alternative to classical methodologies, e.g., based on paper-based instructions or electronic manuals. Nowadays, prototype systems are being studied and proposed by car manufacturers, furniture suppliers, etc. The frameworks and solutions that have been designed and developed in this research, even though primarily focused on maintenance, explore limitations that are shared by every AR application and propose solutions that could be applied in different application domains.

5.1 Workflow Analysis

The workflow analysis to efficiently develop augmented reality markerless applications addresses the most important problems defined in chapter 2, investigating all the steps to design, implement and test an AR application. Finally, two markerless solutions have been compared by a real use case. Performance and robustness of the different systems have been evaluated to identify the best environment-independent tracking solution. The problems found with the 2D image tracking based on render could originate from either the quality of the 3D model or the lack of tracking features by the real object or both of them. Moreover, image tracking based on render should be further investigated as the advantage of using renders relies on the possibility to simulate different texture conditions, such as shadows, dirt, brightness, dust and so on. Future works could include further research on the 3D CAD tracking system if future releases of the Metaio SDK occur (since this tracking solution is only in a beta version), or if other AR framework providing this technology are proposed by its competitors. Moreover, 2D tracking could be improved with a system to better evaluate the quality of the images used as trackables and an algorithm could be developed to dynamically change the recognition threshold on luminosity variation.

5.2 Augmented Maintenance Framework

The main goal is to overcome issues related to the AR content production, thus enabling instructors to easily make and manage training procedures. Results obtained considering a real case study show potential benefits of the AMF. Unskilled people are able to perform a complex task on machine tools by means the AR application; moreover, a client-server architecture allows the instructor both to provide remote assistance to trainees and dynamically change procedures in order to better support students. Future work could be aimed to analyze and measure some indicators such as effort and time needed to train a technician and costs involved in the training process. Moreover, this analysis will be also aimed to investigate different business models related to customer assistance: for some tasks, the augmented reality tool could also replace (or more likely complementary) the traditional assistance program, thus allowing customers to perform maintenance autonomously.

5.3 Interface Creation Framework

This thesis has presented the design and implementation of a framework that supports automatic generation of speech interfaces for controlling VR and AR applications. The generated interfaces include icon-based representations of application functionalities, which are expected to be capable of evoking the relevant commands, thus reducing the cognitive load for the user. The interface creation framework includes visual cues that are automatically selected by means of a semantics-based optimization strategy, which aims at maximizing the match between application functionality and icon description, while limiting possible overlaps between meanings implied by different icons and fostering consistency in terms of graphic style. The automatically-generated interface was tested by first-time users in a case study concerning maintenance of the machine tool, using a wearable AR application. The performance obtained by using this configuration was compared to that achieved in two different scenarios where, in one scenario, the same interface was used by experienced users and, in the other one, a personalized interface was manually created by the users. Objective data, in terms of task completion rate and average number of attempts for activating functionalities, shows that first-time user experience with the automatically generated interface is comparable to that measured with the two reference scenarios. This is confirmed by subjective evaluation. Overall, the automatically-generated interface proved to be more usable and intuitive for first-time users, thanks to the lower cognitive load of activating application functionalities. In the case of personalized interface, intuitiveness of the automatically-generated interface was confirmed by its higher task completion rate and fewer activation attempts. Concerning usability, the interface generated by the proposed framework has the advantage of reducing the number of errors (in terms of false negatives) in activating functionalities. Future research work will be aimed at testing the proposed methodology and the tools developed in different scenarios with new applications and with richer icon sets. Moreover, to improve the quality of semantic mapping and enhance the performance of the speech recognizer, novel strategies will be evolved for extracting context information from icon and functionality descriptions and for expanding the valid voice commands set, preserving the framework's robustness and consistency.

5.4 Speech Interface Robustness

A comparison has been proposed among vocal one switch interfaces based on three different scanning algorithms with a traditional multi-word speech recognition-based interface. The aim is to provide a robust and efficient interface for virtual and augmented reality tasks to be performed hands-free. A bidirectional scanning algorithm has been added to the traditional automatic and inverse ones. The bidirectional algorithm is based on a three-state switch triggered by three words, thus enabling users to select available commands as if they were placed in a circular buffer. In this way, it is possible to improve the robustness performance with respect to a multi-word solution with a limited overhead in completion times. From the user's point of view, although it is the least in terms of robustness, the multi-word solution is preferred in seven cases of eleven; this is due to the fact that the users are asked for pronouncing a lower number of words and they accept a greater error ratio. On the other hand, the BVOS interface is perceived as the best alternative to the MW due to its improved robustness and slightly worse speed. AVOS and IVOS are ranked by the majority of the users as the worst solutions due to the huge gap in latency time with respect to the other two interfaces, although AVOS is the least prone to errors in noisy environments. This study could be extended by considering also other forms of command activation beyond the vocal one; for instance, blinking detection promises to be a robust form of binary activation, which might completely overcome any problem related to noisy environments. Moreover, other scanning algorithms should be investigated when more complex applications have to be managed. The relationship among number of commands to be activated, robustness of the interface and latency time to reach the desired command is still an open problem.

5.5 Conclusion

This chapter summarizes both the positive results obtained through this research and possible future works and areas of investigations. Figure 5.1 illustrates the limitations of the current work and how they could be addressed in future work.

The next chapter will describe a minor research project related to visual analytics that was carried on throughout the PhD.

CHALLENGES	COMPLETION	OPEN PROBLEMS	FUTURE WORKS
POSE TRACKING		CAD Tracking of simple object Recognition of current step completion	Further investigation of tracking algorithms
RECONFIGURABILITY			
CONTENT CREATION		Multiple choices from a given node	Related to tracking capability
INTERFACE DESIGN		Recognition errors could be reduced	Further investigation of speech recognition algorithms and hardware solutions
INTERFACE ROBUSTNESS		Reduce perceived annoyance Reduce errors	Further investigation of one switch algorithms and hardware solutions
TECHNOLOGY ACCEPTANCE		Improve usability of provided solutions	Further investigation of hardware and interface solutions

Fig. 5.1 Open problems and possible future works.

Chapter 6

Multidimensional Data Visualization

Part of the work described in this chapter has also been previously published in [62]

As described in Section 1.1, visualization groups all the techniques adopted to create images, animations or diagrams with the purpose of communicating a message. Data visualization is a branch of the visualization area that focuses on the analysis, design and creation of visual representation of data, that is “information that has been abstracted in some schematic form, including attributes or variables for the units of information” [235]. From a general point of view, the goal of visualization is to communicate a message: in data visualization, the goal is to communicate to the user information in a clear and efficient manner through the usage of graphics, plots or information graphics. Section 6.2 describes in detail the possible usages for data visualization. Moreover, similarly to AR, data visualization poses specific problems related to the interaction between the user and the machine. Data visualization is even more widespread in terms of usage than AR and has application in different fields of use such as science, education, economy, finance, climate, medicine and many others. For these reasons, solutions to open problems posed by this area of research may be very significant.

This chapter presents a brief analysis on multidimensional data visualization. Section 6.1 introduces the key concepts, open problems posed by this branch of visualization and a definition of multidimensional data. The following section describes the possible uses for data visualization, as categorized by Valiati in 1995.

Section 6.3 offers a brief panoramic on the most used techniques for representing multidimensional data. Since this analysis delineates the scatterplot as a renowned visualization tool worthy of further inverstigation, section 6.4 focuses on this tool, presenting different techniques for adapting its usage to multidimensional data. Furthermore, section 6.5 presents a comparison between two different implementations, the ScatterDice and a multidimensional scatterplot implementation. Section 6.7 provides some final considerations and possible future works.

6.1 Introduction

Big Data visualization is a challenging area of research due to the dramatic increment of information produced on a daily base by the information systems and infrastructures all over the world. Datasets grow rapidly, mainly for two reasons: first, they are gathered by an increasing number of cheap, information-catching mobile devices, remote sensing devices (such as satellites and other aerial devices), smartphones, cameras, microphones, software logs, wireless sensor networks and many other devices or source of information [236] [237]. Moreover, the exponential growth of the Internet of Things (IoT) is providing even more services based on data, as they are collected from many different kind of sensors.

As reported by Hilbert & López, the world's technological capacity to store information has roughly doubled every 40 months since the 1980s [238] and IBM estimated that as of 2016, at least 2.5 exabytes of data are generated every day [239]. The most used definition for Big Data was proposed by the Gartner Inc. research group in 2012, as an update of their previous one: "Big Data is high volume, high velocity, and/or high variety information assets that require new forms of processing to enable enhanced decision making, insight discovery and process optimization" [240].

These data are important due to the insight they contain: it can be the trend of the financial markets, the production efficiency of a big company, the most requested holiday destinations or consumer goods. Boyd and Crawford affirm that "There is little doubt that the quantities of data now available are indeed large, but that is not the most relevant characteristic of this new data ecosystem" [241].

6.1.1 Open Problems

Some of the most interesting challenges offered by Big Data include storage, capture, data curation, sharing, transfer, analysis, search, querying, visualization, updating and information privacy. One of the most important problem is how to comprehend and exploit these data so that they provide a cognitive advantage that can led to more conscious choices. The first step forward the understanding of data is the way they are represented: an effective graphical visualization can provide a quick and intuitive way to comprehension. Visual analytics offers a wide plethora of different visualization tools and solutions, but very few of them are specifically tailored to the problems related to Big Data. Moreover, the high variety of information is one of the main problems when applying traditional visualization tools to Big Data, because they are designed to address sets of data with a relative low number of variables for each element of the dataset.

Multidimensional visualizations try to address the challenges of displaying dataset with many variables. This fact suggests two kinds of problems: firstly, the majority of the charts usually adopted to visualize data cannot display more than three dimensions appropriately; secondly, the efficacy of the graphical effects adopted to represent different data dimensions deteriorates when the number of variables to be displayed increases. Visual exploration of multidimensional data is relevant since it helps to find trends, patterns, outliers, and relationships among variables. When visualizing multidimensional data it is possible to map each variable to some graphical entity or attribute.

6.1.2 Multidimensional Data

Datasets with an high number of variables have been defined by Wong and Bergeron as Multidimensional multivariate data [242]: the dataset represents a set of observations X, where the i^{th} element x_i consists of a vector with m variables, $x_i = (x_{i1}, ..., x_{im})$. Each variable m may be independent or dependent on one or more other variables. Independent variables are referred to multidimensional variables and dependent variables are referred to multivariate. This poses another problem, as the user may not know if the data is multidimensional or multivariate, thus if a correlation between the data exists. Eventually, this could be one of the questions the user may want to address when analyzing the dataset with a visualization tool.

6.2 Usage

Different scenarios lead to different tasks when dealing with multidimensional visualization techniques. As defined by Valiati in [243] and further described by Pillat et al. in [244], five major tasks can be considered as objectives a user might want to fulfill when using a visualization tool to display or analyze multidimensional data: identify, determine, compare, infer and locate. Scatterplots can be used to assess all these different tasks and have been applied to data in many different fields of use, such as automotive, finance, pharmacology, environment, weather forecast, telecommunication, food and many others.

6.2.1 Identify

This task refers to any action of finding, discovering or estimating visually:

- properties like symmetrical or asymmetrical distribution, values or dispersion;
- correlation, data dependency or independency;
- similarities or differences;
- clusters as a result of similarity, continuity, proximity or closed shapes;
- thresholds, patterns, data variation.

The Identify task takes place anytime the user analyzes the chart with the purpose of finding, estimating or discovering new information about the data. The task ends when the user finds the information he/she was looking for or the current goal changes. Figure 6.1 shows an example of scatterplot that clearly suggests a linear correlation among the displayed variables.

6.2.2 Determine

This task corresponds to the action of calculating, defining or precisely designating values such as:

- mean, median, variance, standard deviation, amplitude, percentile;

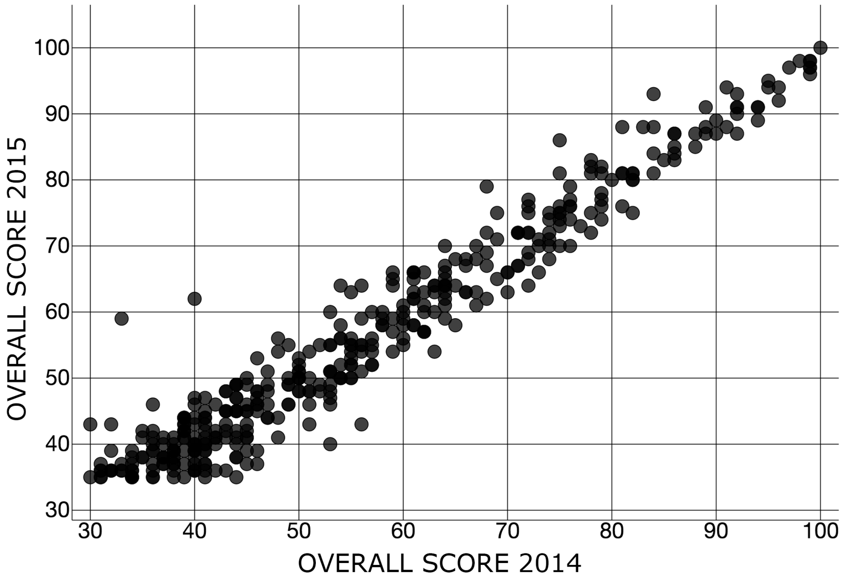

Fig. 6.1 This scatterplot suggests a correlation between the two displayed variables.

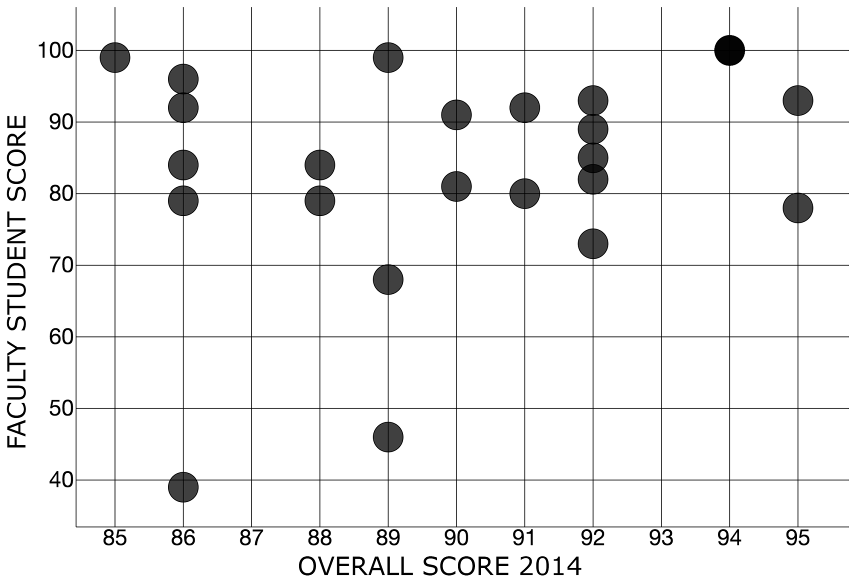

Fig. 6.2 A scatterplot visualization that simplify the computation of the mean.

- sum, differences, proportions;
- correlation coefficients, probabilities or other statistics such as hypotheses test.

This task begins when the user needs to calculate a specific value and ends up when the calculation is completed. Figure 6.2 shows a scatterplot that allows the user to

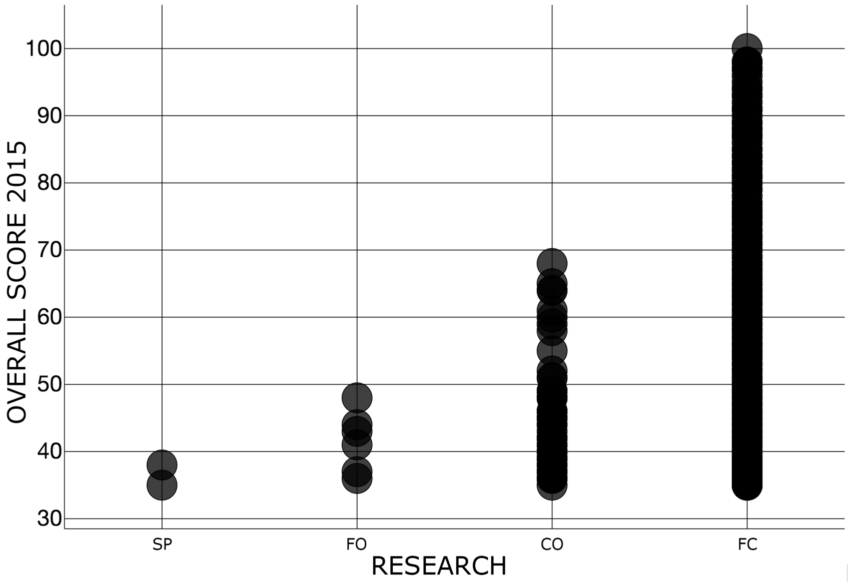

Fig. 6.3 A scatterplot configuration that enhances comparison.

derive the precise value of each point in order to compute precise calculations such as the mean value.

6.2.3 Compare

This task takes place when the user wants to compare data that have been previously identified, located, visualized or determined. The user may compare data to analyze dimensions, data items, clusters, properties, proportion, values, locations and distances or visual characteristics. The Compare task is an analytic task the user performs specifically if he/she compares data items displayed in the graphical visualization. Figure 6.3 shows a scatterplot configuration that enhances the comparison task.

6.2.4 Infer

This task refers to the action of inferring knowledge from the visualized information, such as defining hypotheses, rules, probabilities or trends, attributes of cause and effect. This task usually takes place after determining, identifying or comparing information and it is performed as part of the mechanism of data analysis, thus it may not be completed at once, requiring consecutive applications of the other

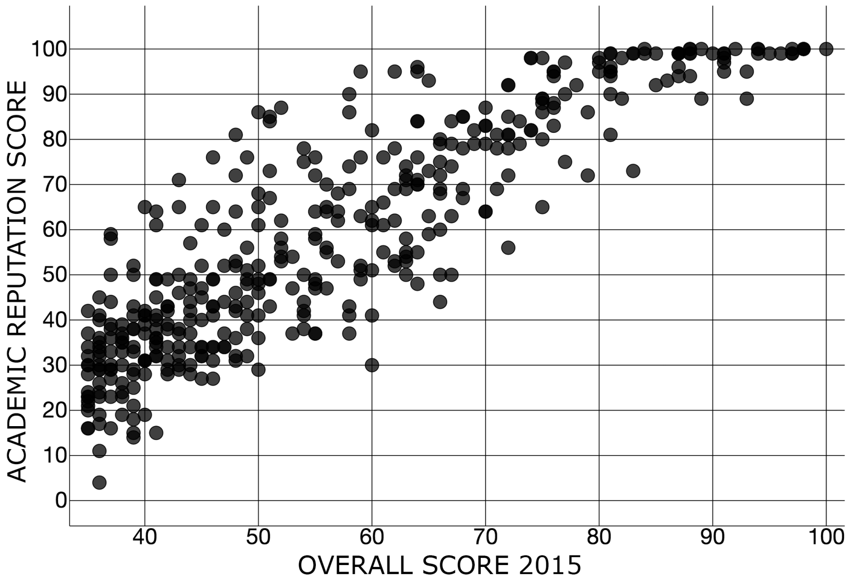

Fig. 6.4 A scatterplot that enables the user to infer a hypothesis.

visualization tasks. Analyzing Figure 6.4 it is possible to infer a hypothesis, e.g. that the y variable is the cause of the trend of the data.

6.2.5 Locate

This task refers to the actions of searching and finding information in the graphic representation: they can be data points, values, distances, clusters, properties or other visual characteristics. The task begins when the user starts examining the visual representation and finishes when he/she recognizes the desired information. Figure 6.5 shows a scatterplot visualization that enhances the identification of outliers.

6.3 Related Works

The traditional two-dimensional point and line plots are among the most commonly used visualization techniques for data with lower number of variables. From the evolution and modification of these two main techniques, one derives different types of representations related to multidimensional visualization. Different approaches to different visualization tasks and problems have led through the years to many kind

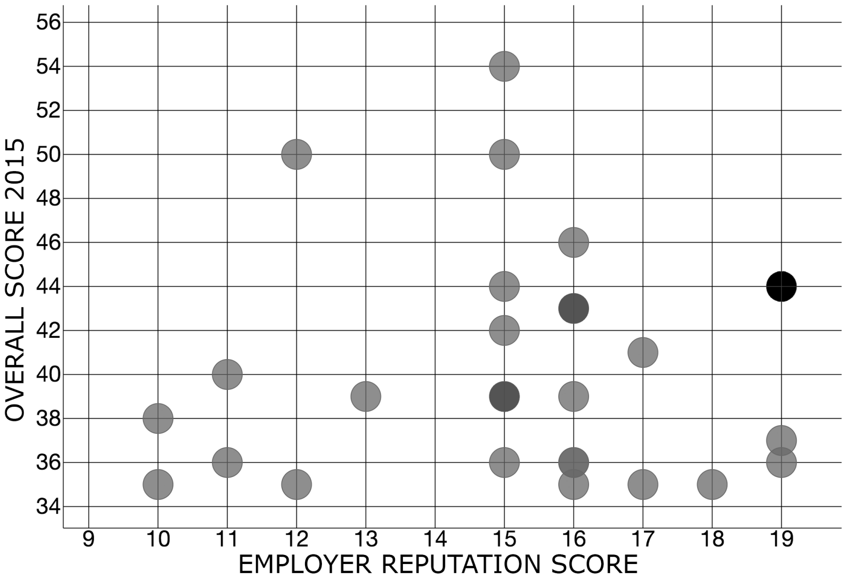

Fig. 6.5 A scatterplot visualization that enhances the identification of outliers.

of visualization techniques and graphic tools. This section briefly describes the most important ones when dealing with multidimensional data.

6.3.1 Small Multiplies

Edward Tufte was the first to introduce the concept of small multiples, describing them as: "Illustrations of postage-stamp size are indexed by category or a label, sequenced over time like the frames of a movie, or ordered by a quantitative variable not used in the single image itself" [245]. Figure 6.6a shows an example of small multiplies. Small multiples allow to visualize multidimensional data without compressing all the information in a single, overly complex chart. Moreover, they offer some valuable features: overplotting, that is the display of too many variables with the risk of confusing the audience, is avoided; easiness of use, since the reader can learn to interpret a single chart and then he can apply the same method to the rest of the charts; finally, small multiples simplify comparisons across variables and help the user to identify patterns in the charts. Limitations include: the proportions may make it difficult for the user to easily understand the insights of a single charts; if a logic is missing in the placement of the charts, the user may focus his attention only on the understanding of a specific order instead of the subject inside the chart.

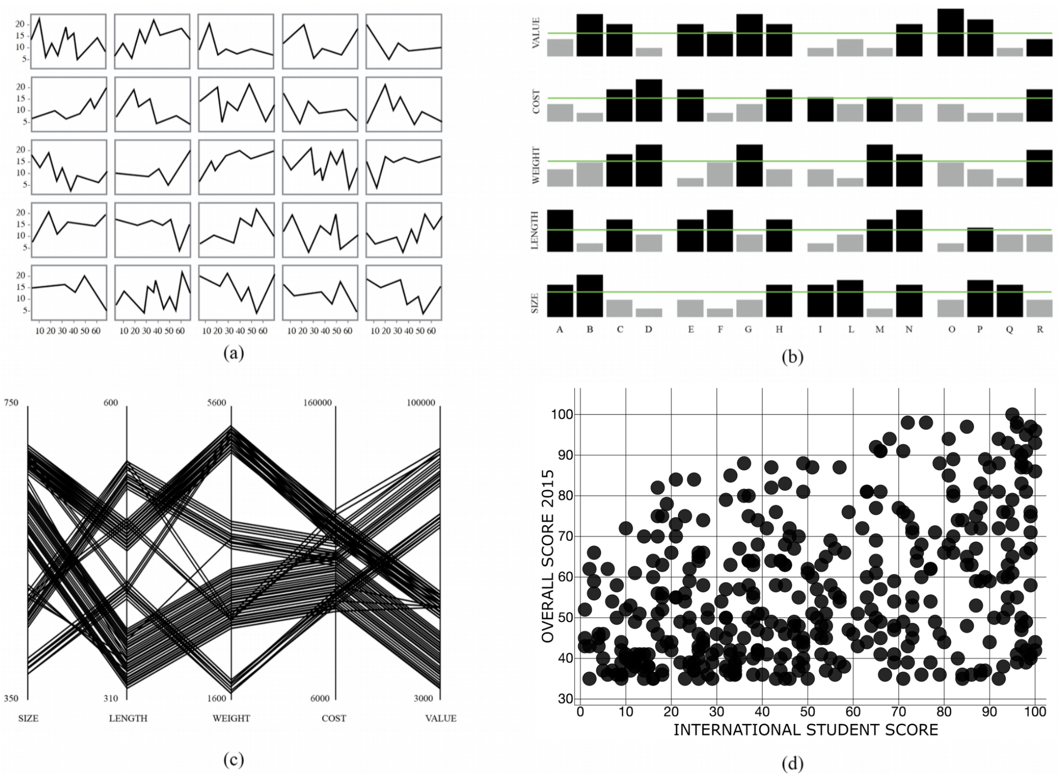

Fig. 6.6 Small Multiplies (a), Permutation Matrix (b), Parallel Coordinates (c) and Scatterplot Matrix (d).

6.3.2 Permutation Matrix

Bertin describes the permutation matrix as a peculiar type of bar graphs [246]. Usually, a bar graph can display only one-dimensional data, with the heights of the bars representing the data values. In a permutation matrix, multiple bar graphs are displayed, one for each data dimension. Figure 6.6b shows an example of permutation matrix. The technique consist of representing rows of data as bar charts and sorting them. The goal of the permutations is to form patterns, typically to place the large values on the diagonal of the matrix, thereby clustering similar cases with their representative variables [247]. The horizontal axes of each bar graphs represent the same information. Moreover, to enhance data comparison, all data values above average are coloured white and all data values below average are coloured black. Finally, the average value of each dimension is represented by a a green dashed line plotted over the data. Some implementations of the permutation matrix allow to sort rows and columns to simplify pattern identification [248]. Similarly to the previous techniques, the proportions may make it difficult for the user to easily understand the insights of a single charts. Moreover, as other ‘matrix’ techniques, analyzing all the plots may require a lot of time, depending on the number of variables. Thus, this peculiar visualization technique is most suited for comparison tasks.

6.3.3 Parallel Coordinates

Parallel coordinates, representing multidimensional data using lines, were formalized for the first time by Inselberg [249]. Each data dimension is represented by one parallel axis, with the maximum and minimum values of each dimension scaled to the upper and lower points on a vertical axis. An n-dimensional dataset is represented by polylines that cross each axis at a different position, depeding on the value of that element for that dimension. Figure 6.6c shows an example of parallel coordinates. This visualization technique is expecially suitable for detecting and characterizing outliers, as well as comparing elements and detecting relationships among variables. One of the most important limitation of this technique is that the dimension it can express are limited by the length or width of the screen. Moreover, when representing huge datasets, overplotting may drastically reduce the understandability of the visualization.

6.3.4 Scatterplots

A scatterplot (Figure 6.6d) is a chart or mathematical diagram displaying a set of data as a collection of points using Cartesian coordinates, usually defined by horizontal and vertical axes. Each point on the chart represents two variables, x and y, calculated independently to form bivariate pairs (x_i, y_i). A functional relation between x and y is not necessary. The purpose of a scatterplot is to reveal (if existing) the relation between the displayed variables.

Scatterplots may be considered, among the different types of data visual representations, as one of the most useful and versatile, especially in statistics. According to [250], the term first appeared as Scatter Diagram in a 1906 article in Biometrika, “On the Relation Between the Symmetry of the Egg and the Symmetry of the Embryo in the Frog (Rana Temporaria)” by J. W. Jenkinson. However, the term only came into wide use in the 1920s when it began to appear in textbooks, e.g. F. C. Mills, Statistical Methods of 1925. The Oxford English Dictionary gives the following quotation from Mills: "The equation to a straight line, fitted by the method of least squares to the points on the scatter diagram, will express mathematically the average relationship between these two variables".

Scatterplots are mainly appreciated for their ability to reveal nonlinear relationships between variables. Moreover, scatterplots are typically used to identify correlations between variables, with a certain confidence interval. Another usage for the scatterplot is to compare similar datasets. Since the main problem of multidimensional data is to correctly understand and analyze them, pointing out relationships, patterns or outliers, a scatterplot provides a suitable visualization tool for multidimensional data due to its intrinsic features.

6.3.5 Remarks

The scatterplot visualization diagram is considered one of the most functional among the variety of data visual representations, due to its relative simplicity in comparison to other multidimensional visualization techniques [251]. Moreover, multidimensional visualization tools that feature scatterplots, such as GGobi [252], Tableau/Polaris [253] and XmdvTool [254], usually allow the user to map data dimensions to additional graphical properties such as point color, shape, and size. For this reason, the scatterplot has been chosen as the reference tool for this analysis. Scatterplots can be used to assess all the different tasks defined in 6.2 and have been applied to data in many different fields of use, such as automotive, finance, pharmacology, environment, weather forecast, telecommunication, food and many others.

Moreover, the analysis of the existing visualization tools highlighted a number of utilities that a multidimensional visualization tool should comprehend to enhance the user experience, such as:

- the tool should allow the user to input numerical data as well as alfabetical and discrete variables;
- the tools should enable the user to define filters in order to simplify the visual exploration of the dataset;
- a zoom function should allow the user to better evaluate and compare points, especially when occlusion occurs;

- users should be enabled to map the data dimensions to the available graphical effects as they please;
- instruments for interactive refinement of the dataset should be available.

6.4 Scatterplots for multidimensional Data visualizations

The main problem when using the scatterplot to visualize multidimensional data is that its basic version is limited to only two variables, thus making it difficult to correctly visualize and analyze all the data. In order to overcome this problem, different solutions have been proposed through the years to enhance the scatterplot. Section 6.4.1 describes the most used techniques for increasing the number of dimensions displayable through a scatteplot through additional graphical effects. Sections 6.4.2, 6.4.3 and 6.4.4 describes three different evolutions of the scatterplot designed to enhance its usability when dealing with multidimensional data.

6.4.1 Adding Dimensions

Even if the basic scatterplot may display only two variables, various techniques have been researched and adopted through the decades to increase the dimensionality of scatterplots by one, two, or even several additional dimensions. A bidimensional planar scatterplot of two variables X and Y can display additional variables by correlating them to one or more graphical features of the plotted points.

Color

One approach is to show a third dimension through a color map. Colored points on a scatterplot may suggest similarity among values of the same dataset or correspondence among points of different datasets. Moreover, this correlation may be perceived without drawing any connecting line. This technique is particularly powerful since it could also be used to link together an arbitrary number of scatterplots, both different or complementary, such as in the case of a scatterplot matrix, without cluttering or visibly degrading any of them. This solution can increase significantly

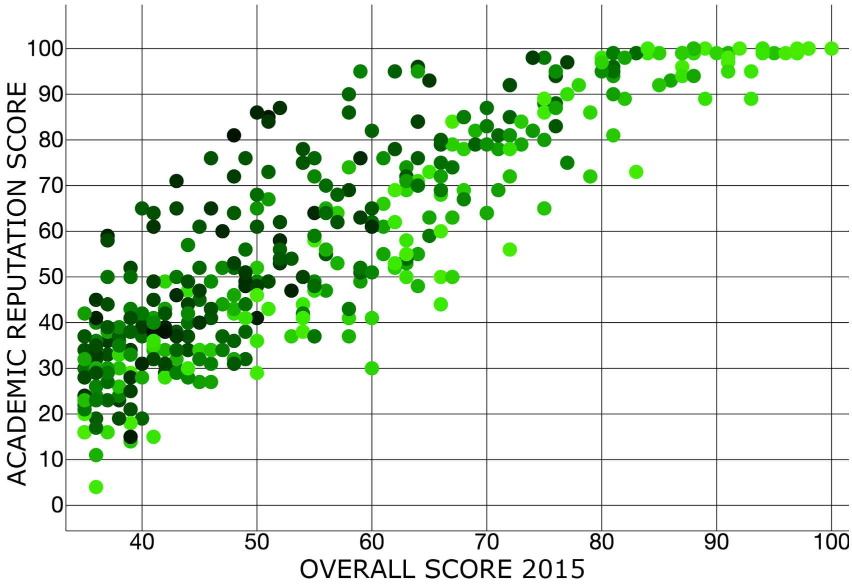

Fig. 6.7 A scatterplot with an additional variable visualized as color.

the effectiveness of such visualization with respect to the sum of the individual unlinked scatterplots. Colors can also be used to enhance the perception of a variable already displayed by another effect (such as an axis). Figure 6.7 shows a scatterplot that displays an additional variable through colors.

Size

A further option to provide an additional dimension to the scatterplot is to vary the size of the points. This option may lead to occlusion problems if the plot does not provide proper scaling on the two axis. Figure 6.8 shows a scatterplot with a variable mapped on the size of the points.

Shape

Another approach is to add a third dimension changing the shape of the points. Instead of using only points, each element of the dataset could be drawn as different kinds of glyphs depending on a third variable. This option leads to further possibilities in terms of the paradigm used to choose the shape. One option is to display the points as ‘flowers’, relating the variable to the number of ‘petals’ to display. Another option is to display polygons and relating the number of sides to the variable.

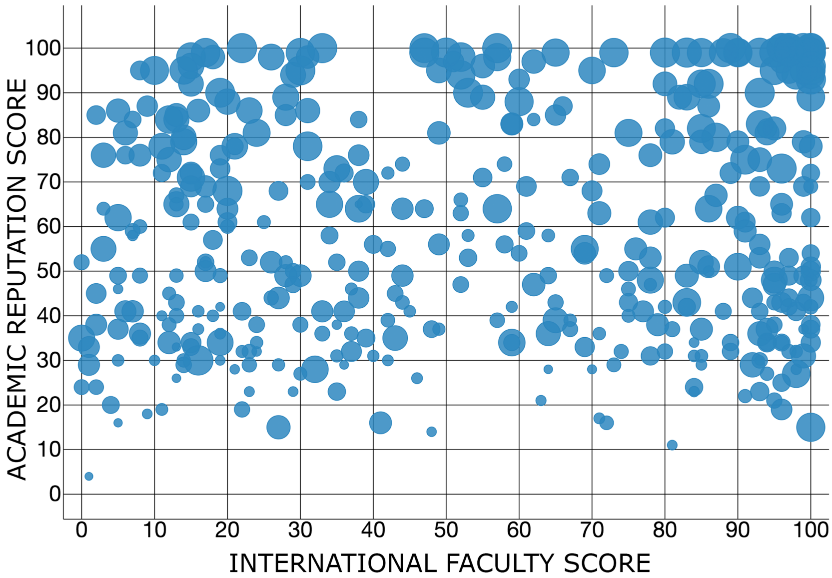

Fig. 6.8 A scatterplot with a variable mapped on the size of the points.

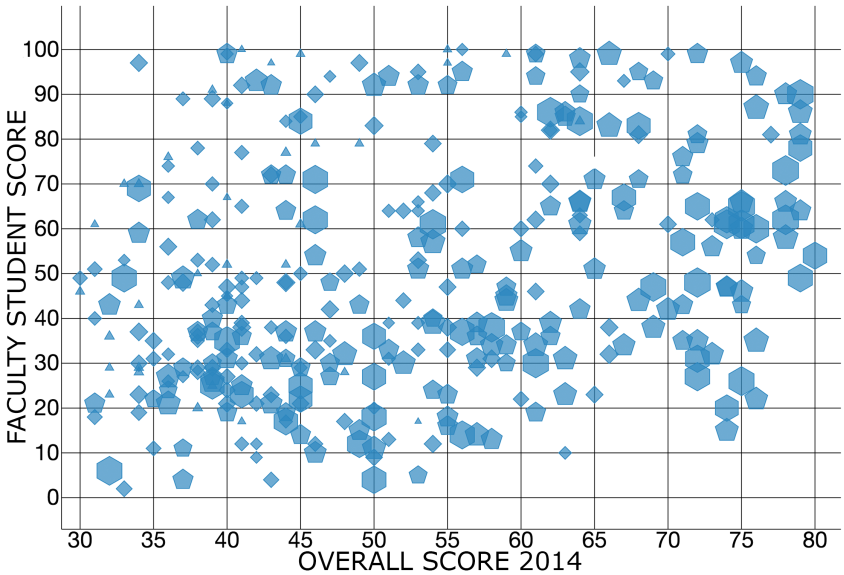

Fig. 6.9 A scatterplot that uses the shape of the points to display additional information.

Moreover, various glyphs, clearly distinct among them, could be used to represent different datasets. Figure 6.9 shows a scatterplot that uses the shape of the points to display additional information.

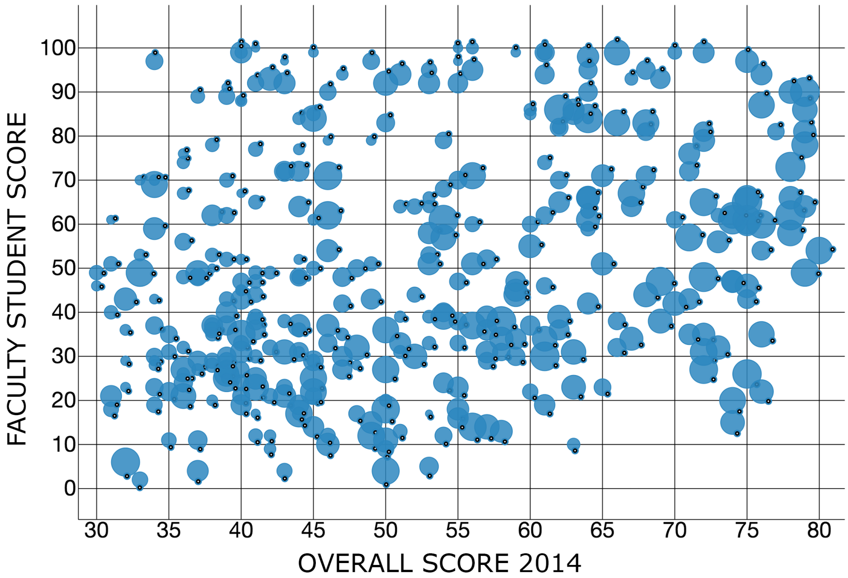

Fig. 6.10 A scatterplot that displays an additional variable through orientation.

Orientation

Another possibility when displaying points as shapes is to represent a third dimension changing the orientation of the shape. Usually a dot or line is drawn orthogonally to the perimeter of the shape to better identify the reference point for the orientation. Figure 6.10 shows a scatterplot that displays an additional variable through the orientation of the points.

Error Bars

The uncertainty is the variability related to a specific variable of the dataset for each point. It provides a generic idea of how precise the measurement of the reported value is, or how far from the recorded value the real value might be. This information is usually reported through error bars if it is related to a variable mapped on the x or y axis (or both). Figure 6.11 shows three examples of error bars. Error bars require additional space around the points to be correctly displayed due to the chance of overlapping between points. For this reason they are usually adopted only if the points of the scatterplots are very scattered and occlusion does not occur. Otherwise, due to the space needed to draw the bars, their use would greatly affect the understandability of the representation. As a result, the use of error bars limits the

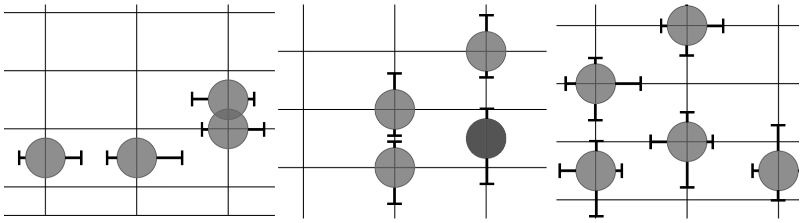

Fig. 6.11 Three examples of error bars.

number of different graphical effects that could be combined on the same scatterplot and should be avoided when displaying more than three or four variables.

Adding More Dimensions Concurrently

It is possible to use simultaneously more than one of these techniques, independently, to obtain even high visual dimensionality. Figure 6.12 shows an example of such a scatterplot. However, this is recommended only if the graphical effects are clearly distinguishable, otherwise the visual clarity and benefits of displaying more dimensions at the same time will promptly worsen. Many studies, like the one by [255], have been carried out to understand how visualization design can benefit from taking into consideration perception, as different assignments of visual encoding variables such as color, shape and size could strongly affect how viewers understand data.

Dynamic Visualizations

Even if scatterplots are typically used to display static data, nevertheless they can be very useful when applied to display data that could change dynamically, moreover if the change may be controlled by the user. More complex graphical effects such as animation may be adopted in this case to enhance the comprehension of data as they change over time. This is the case of data characterized by one or more time-related variables, such as stocks values in finance or weather conditions in forecasting.

6.4.2 Scatterplot Matrix

The simplest approach to adapt the scatterplot to multidimensional data is to produce a series of scatterplots for each pair of variables and display them together on a single screen or page. This visualization technique is called Scatterplot Matrix and for k

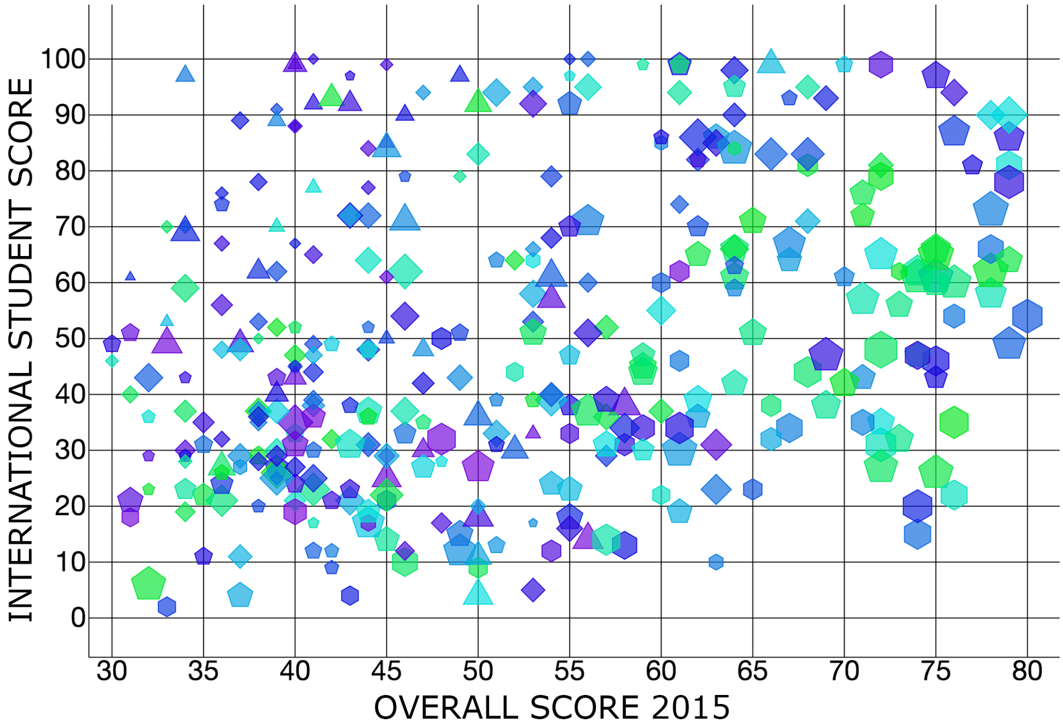

Fig. 6.12 A scatterplot that displays multiple variables through different effects.

variables it requires $\frac{k(k-1)}{2}$ pairs and therefore scatterplots. Unfortunately, this solution presents a major problem: analyzing all the scatterplots may require a lot of time, depending on the number of variables, thus this solution is not optimal when dealing with time-related tasks. To overcome this problem, different visualization techniques may be adopted to interact with the dataset and simplify data comprehension. Figure 6.13 shows an example of scatterplot matrix.

Brushing

Brushing is the action of selecting a subset of the points displayed on the scatterplot. Four brushing operations have been defined [256]: highlight, shadow highlight, delete, and label. To perform these operations, it is necessary to resize a rectangle, called the brush, over one of the scatterplots. The corresponding points on each different scatterplot are then affected by the chosen operation. The brush can be moved to different regions of the scatterplot by moving the mouse. At any time, the user can stop the brushing operation, change the shape of the brush or the chosen operation and then resume the brushing.

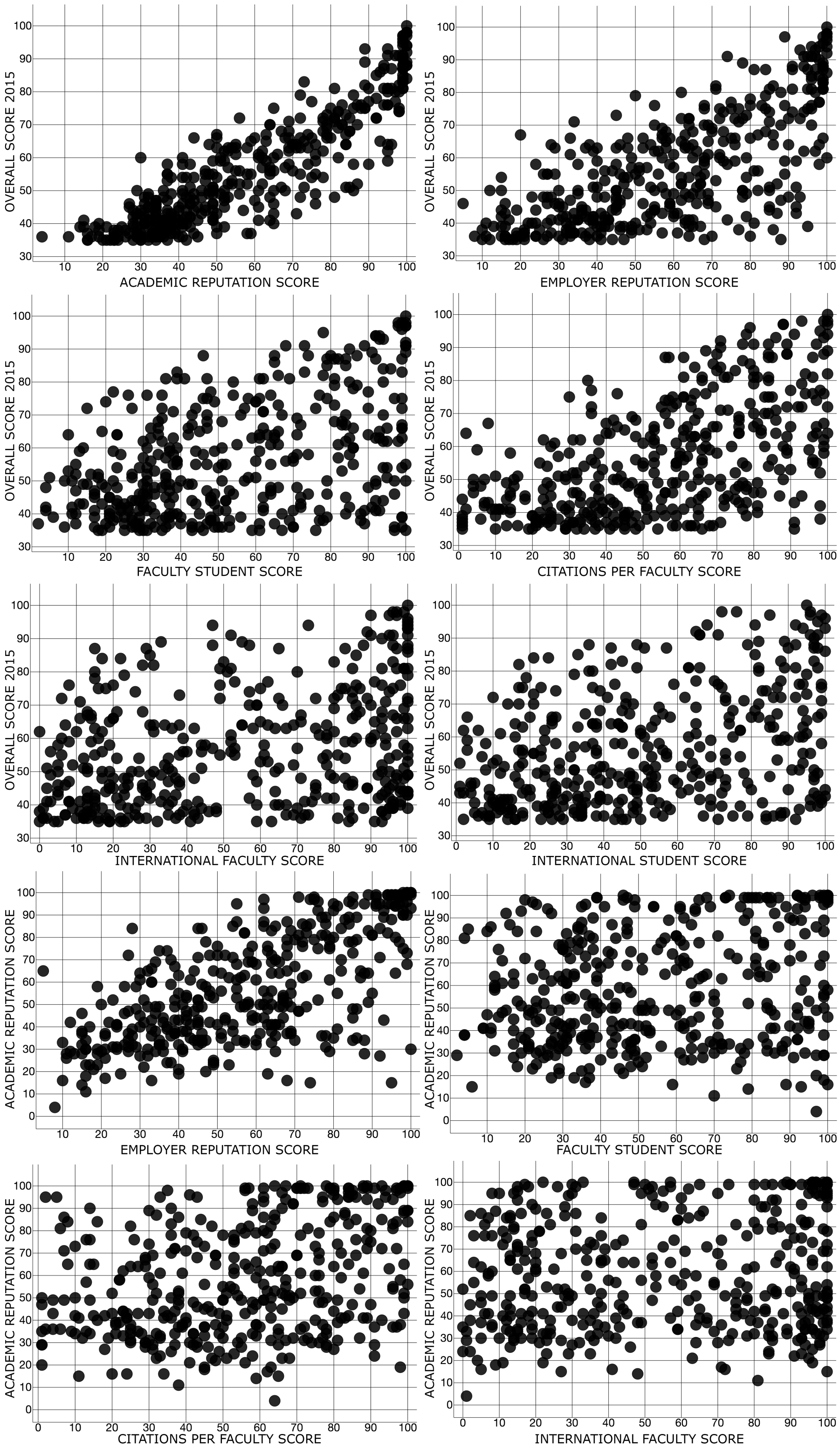

Fig. 6.13 An example of scatterplot matrix.

Dimension Reordering

One of the problems when dealing with scatterplot matrixes is to simplify the understandability of the data. One possibility is to change the way the scatterplots are displayed and ordered to enhance the presence of clusters, patterns or trends. Different approaches have been investigated and adopted, such as the systematic dimension reordering approach of [257] where similar dimensions in a multidimensional dataset are placed next to each other. Using a scatterplot matrix it is possible to order independently rows and columns. In the systematic dimension reordering approach, similarity are displayed on the column and dissimilarity on the row order.

6.4.3 3D Scatterplot

Another way to display multidimensional data through scatterplots consists in adopting a 3D visualization. 3D scatterplots exploit the third dimension, representing three data dimensions on the x, y and z coordinates, in a three-dimensional space. The third dimension allows the user to interact with the scatterplot to change the viewport (with two or three degrees of freedom). Hypothetically, more coordinates could be added to the model, leading to an n-dimensional spatial representation. Since 3D scatterplots are represented on displays as 2D images, the 3D representation needs to provide useful hints to properly display depth and avoid occlusions or misinterpretations of data. Occlusions can be addressed also in 2D representations by using another data dimension for depth sorting. The latter can also be compared to a full 3D scatterplot where the only difference is the missing rotational interaction in 3D. This mapping also requires three axis: two for spatial positions and one for sorting. 3D scatterplots make it possible to obtain more flexibility in the data mapping simply avoiding to fix certain data dimensions to only certain specific scatterplot axis: this could be obtained allowing the user to exchange the dimensions mapped on each axis, either by swapping the dimension of one or two axis or by manipulation of dimensions. 3D scatterplots may also consist of more complex versions, including additional graphical effects (color, size, orientation, shape, etc.) to represent additional information related to the displayed data, guideways (reference lines from the data up to some reference points) and combinations of scatter data with additional objects as fit surfaces. A common application of the 3D scatterplot is to show both experimental and theoretically-adjusted data in order to be able to determine the

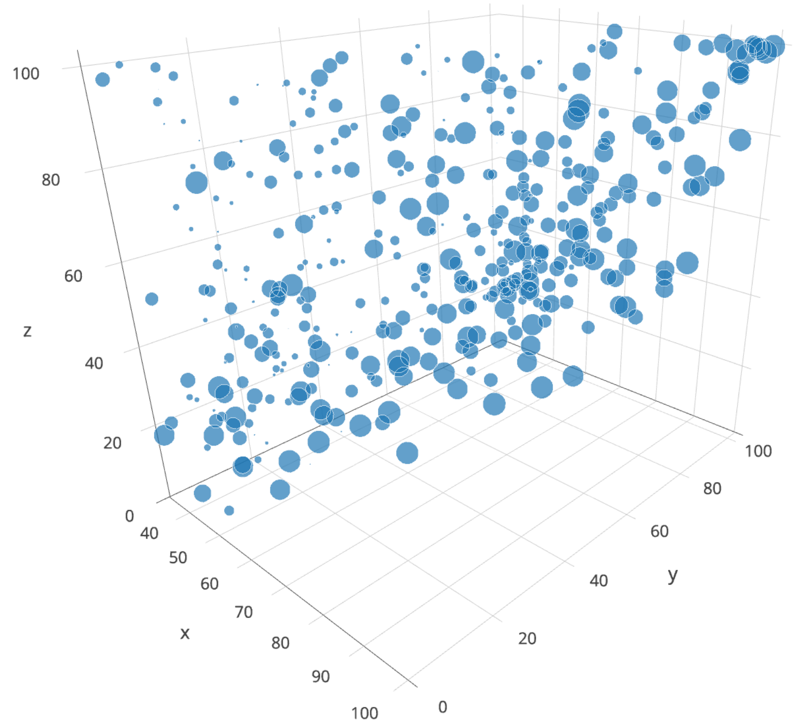

Fig. 6.14 A 3D scatterplot displaying an additional variable through size.

points of agreement. In Figure 6.14, a scatterplot can be observed in three dimensions that makes use of the size of the spheres to map an additional attribute. Overall, 3D scatterplots have certain advantages and limitations with respect to 2D models, as depicted by [258].

Advantages

In a 3D scatterplot, maintaining the same density of points as in a 2D scatterplot involves increasing the number of experimental data to be displayed (larger sample space). If the number of points of the initial 2D scatterplot is maintained, there is a greater discrimination of the relations among variables, since a characteristic is added to the data. The use of 3D scatterplots with volume visualizations for the glyphs to represent the data provides the possibility of using procedural techniques to generate the forms [259]. These techniques allow the user to increase the number of dimensions of the data to be shown by exploiting the shape of the glyphs, thus taking advantage of the pre-attentive ability of the human visual system to discriminate forms. To obtain the best result from a 3D scatterplot, it is necessary to achieve an efficient attribute mapping and to provide the necessary interaction tools to navigate and examine the data: these requirements enhance the expressive power of a 3D scatterplot and allow the user to analyze complex relationships among multiple variables.

Limitations

It is not advisable to abuse multidimensionality if it is not absolutely necessary and the result is not visually illustrative. Moving information representations from 2-dimension to 3-dimension is not a simple task, since the extra dimension may greatly affect how information can be presented and interpreted. The visualization must make an efficient use of the additional dimension and avoid that the new representation is misinterpreted by the user as a consequence of an inappropriate mapping. Special consideration must be given to the perception of spatial distance. The size of the objects can cause the user to not perceive the correct perspective of the information shown: it is difficult to discriminate among the different depths of the objects and to address this problem it is necessary to provide the appropriate interactions tools. A disadvantage arising from the use of three-dimensional objects is the occlusion, which occurs when one object covers another or occupies the same spatial position for two coordinates in the 3D representation. This type of problem occurs mainly when the density of data items to be displayed is large, or when simply a very large object is positioned in front of smaller objects.

6.4.4 ScatterDice

ScatterDice is a visualization technique designed to explore large and multidimensional datasets by navigation in data dimension space using 2D scatterplots and a matrix of scatterplots [260]. For each dimension in the dataset, the ScatterDice creates one scatterplot per every combination of dimensions and arrange them in a large scatterplot matrix, used as an overview of the dataset. Next to the matrix, a standard bidimensional scatterplot is displayed, representing the currently seleceted scatterplot from the matrix. 3D rotations visually represent the transitions from one scatterplot to another. Furthermore, the user may perform visual queries using bounding volumes and iteratively refine a query changing viewpoints. Figure 6.15 shows an example of ScatterDice visualization.

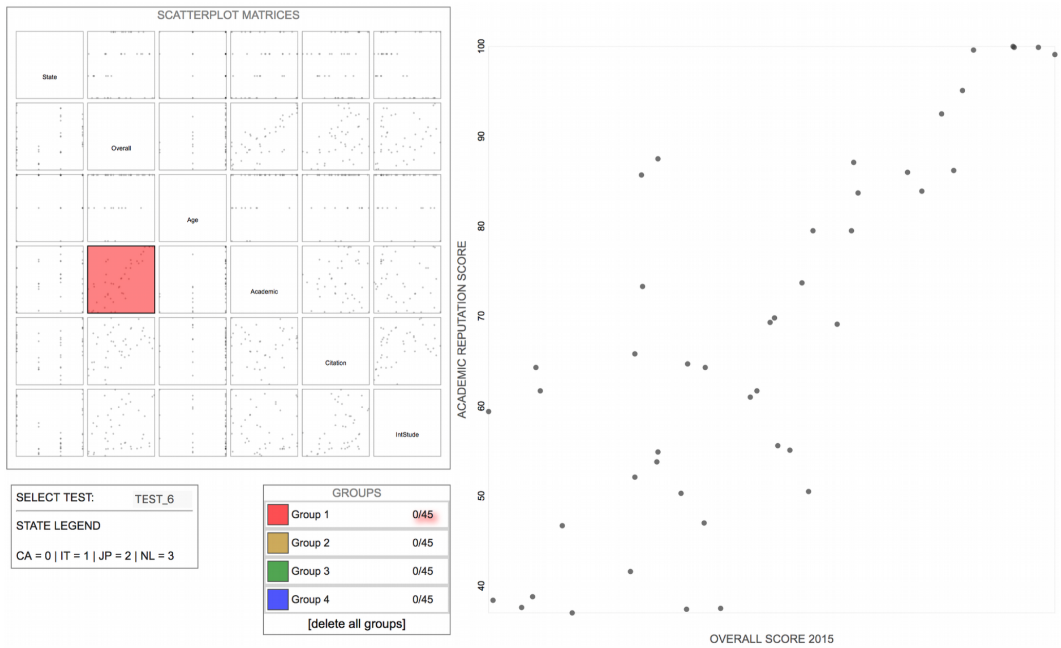

Fig. 6.15 ScatterDice.

6.4.5 Remarks

The reason behind such a various enumeration of scatterplot solutions is that none of them could be considered the best version: each implementation could be less or more useful depending on the specific task the user intends to solve. Eventually, more than one kind of scatterplot should be used for the same dataset to address different tasks. Overall, a simple classification could distinguish among 3D scatterplots, scatterplot matrices and standard scatterplots with additional dimensions. 3D scatterplots are more useful when dealing with a huge amount of data with a dense distribution on the x and y axis, allowing the user a better analysis through spatial navigation. Scatterplot matrices are more useful when the task is to search for correlations between two variables of the dataset: each scatterplot of the matrix may display two variables, and the user just need to analyze them all, one by one. For other tasks, the best solution is adding dimensions to the standard scatterplot, as different graphical effects provide a better insight on the data depending on visual perception criteria, as investigated by [255] and many others.

6.5 Multidimensional Scatterplot Design and Implementation

The analysis detailed in the previous section depicts the ScatterDice as one interesting evolution of the scatterplot. At the same time, a classical scatterplot enhanced with additional graphical effects could be more generalist and more flexible considering the different usage for visualization tools described in section 6.2. For this reasons, it could be interesting to compare the ScatterDice with an implementation of the scatterplot that could visualize more dimensions than the two dimensions of the basic version and possibly even more than four since many common solutions display up to four dimensions through the usage of color and size effects. In order to carry out this comparison, an implementation of a multidimensional scatterplot has been developed. Section 6.5 describes the design and implementation of a multidimensional scatterplot that can visualize up to 8 dimensions at the same time through different graphical effects. Section 6.6 describes the preliminary tests that have been performed and some evaluations that could guide further investigations in this area of research.

Multidimensional Scatterplot Design and Implementation

The multidimensional Scatterplot was developed as a web application using HTML, CSS and Javascript. The D3.js [261] Javascript library have been used to both load and manage the input dataset and to provide the visualization. The D3.js library is designed to provide efficient manipulation of documents based on data, supporting large datasets and dynamic behaviors for interaction and animation. Moreover, D3.js make use of SVG to graphically display the data on the HTML canvas. The algorithm behind the multidimensional Scatterplot fulfills two core steps: loading the dataset from the source file and displaying the dataset on the screen. The first phase starts loading the list of available parameters for the dataset and the type (numeric, alfabetical or discrete) is defined. Then, the dataset is loaded from a comma separated values file (.csv). At this point, when a graphical representation is requested, the following steps are performed: first, for each parameter mapped on a graphical effect, the minimum and maximum values are computed to define the input range. Based on this range, the corresponding scale function is defined: for each value in input inside the given range, the proper value for the corresponding graphical effect is provided as an output. Secondly, if any filter has been defined, it is applied on the dataset. Finally, for each element in the dataset, the drawing function computes the position and the graphical representation of the values, as well as the tooltip. Moreover, if the size effect is mapped on a parameter, then the dataset is sorted from the biggest shape to the smallest one. This allows to print on the screen the smaller elements on top of the bigger ones, in order to reduce occlusion problems. In order to allow the visualization of eight variables at the same time, eight graphical effect have been defined for the scatterplot. In addition to the position on the x/y axis and to the four graphical effects described in the previous section, two other effects have been created: first, adding a texture and changing its grain it is another option to enanche the chart with an additional variable. Second, varying the tickness of the border of each shape in order to add an additional dimension.

Interface

The default configuration maps two parameters on the x and y axis, with a standard size and color for all the shapes displayed on the chart. The shapes are displayed with a 30% value of transparency in order to simplify the understandability of the chart and easily identify overlapped shapes or occlusion. All the shapes have a solid border of 1 px to distinctly identify each shape on the chart. As displayed in Figure 6.16, the proposed tool, in addition to the scatterplot representation of the dataset, assigns a vertical section on the left of the viewport for the available configuration panels. Moreover, two range bars displayed on the sides of the scatterplot allows the user to zoom on the x and y axis of the chart.

The settings panel provides two drop down menus, one for the list of available visual effects and another one for the parameters. The list of visual effects includes: position on the x or y axis, size of the shape, number of sides, color of the shape, tickness of the border, orientation and texture. The other drop down menu lists all the parameters that are available for the current dataset. Through this menu, the user can choose how to map one or more parameters on the different visual effects. Each visual effect, when selected, allows the user to define the default value (except for the x and y axis), such as the default color of the shapes, the default size and so on. Likewise, when a parameter is mapped on the selected visual effect, it is possible to set the corresponding effect for the minimum and maximum values of the dataset, e.g. the size of the shape for the minimum and maximum values of the dataset. Moreover,

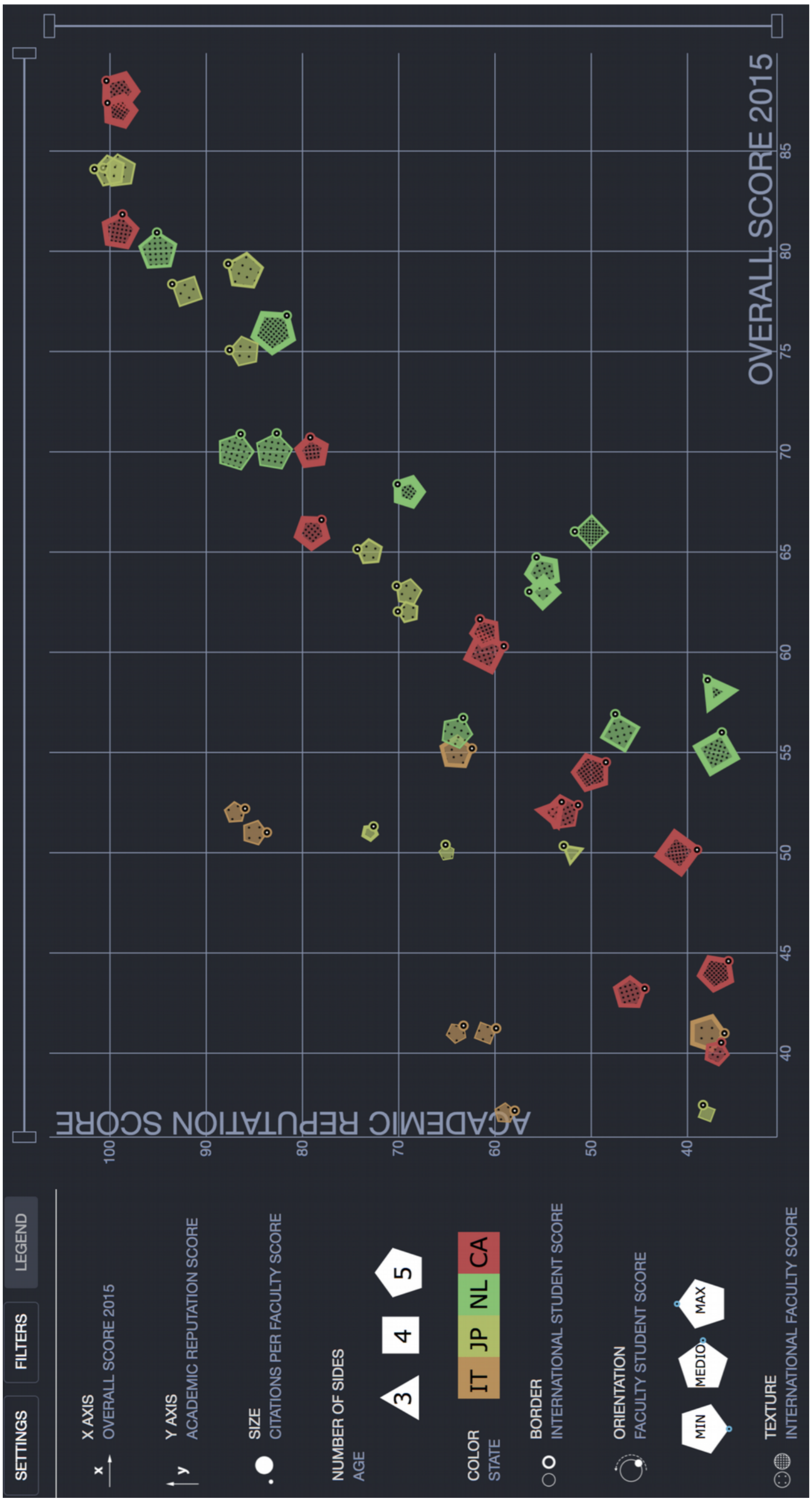

Fig. 6.16 Multidimensional scatterplot.

when a parameter is mapped on a visual effect, other two flag buttons are available for the user: the first one allows the user to switch the graphical representation of the minimum and maximum value; the second one allows the user to apply the visual effect with an absolute or relative scale. Absolute scale means that even if the user zoom in the chart through the side bars, the effect on each shape of the chart will be the same. Relative scale means that the scale for the current graphical effect is applied only on the visible shapes. This implies a minimum and maximum value will be always displayed on the screen. This option, that should be avoided when looking out for absolute value, became useful when performing a comparison task, since it is easier to distinguish between shapes with almost identical values, enhancing their differencies through the graphical effect. The filter panel allows the user to defined one or more filter. It is possible to define more than one filter for each parameters. The interface allows to choose between range filter and list filter. A range filter define a set of values that should be included of excluded through a minimum and maximum values, that delimit the range. The delimiters values may be included or excluded from the selection, singularly or both. This kind of filter is often used when dealing with numerical values. A list filter allow to define a specific list of values that should be included or excluded from the visualization. This filter is usually adopted for alfabethical or discrete parameters. The legend panel resumes all the selected mappings with a miniaturized representation of the graphical effect and the name of the current parameter mapped to it.

6.6 Tests

To evaluate the performance and usability of the proposed tool, a use case has been defined following the methodology described by Valiati et al. in [243] in order to address all the different visualization tasks available. In the examples discussed through the paper, the dataset from QS World University Ranking has been used [262]. The QS World University Ranking is based on six attributes: academic reputation (weighted 40%), employer reputation (10%), faculty/student ratio (20%), citations (20%), international faculty ratio (5%), and international student ratio (5%). These attributes define the overall score.

6.6.1 Use Case

The test consisted of three question that required the user to perform a research in the dataset through subsequent refinements. For each question, the user had to choose among four possible answer. To successfully perform the given task, the user has to locate and compare items, identify differences between them, determine specific values and infer the (possibly) right answer. For each question, the user had to answer using firstly the multidimensional Scatterplot an secondly the ScatterDice. Some settings such as the filter panel were disabled in order to provide a more realistic comparison between the two tools. At the end of the test, the user had to compile an usability questionnaire to provide a qualitative evaluation of the two tools.

6.6.2 Usability Evaluation

For the qualitative study eleven participants (9 male and 2 female) have been recruited; their age ranged between 22 and 34. They were all students with a background in computer science or engineer. In a pilot study with two additional participants, it has been ensured that the overall process runs smoothly and that the tasks are easy to understand. Prior to the actual study, the participants have been checked for color blindness, and in one case it was verified that the color palette was reliable to correctly perform the test. Afterwards, the multidimensional Scatterplot, the ScatterDice tools and the study dataset were introduced to the users. The proposed task had different levels of complexity, with the first one easier, the second one of average difficulty and the third one very difficult. Moreover, the number of visual effects displayed increases in each task, with the first one based on 6 parameters, the second one on 7 and the last one on 8 respectively. The first question was proposed with the purpose of letting the participants familiarize themselves with the software. Moreover, the users were invited to ask questions concerning the concept and interactions. Overall, the introduction and warm-up phase took about 10 minutes per subject. Participants were advised to ‘think aloud’ during the study. In addition to measuring task completion time for the answers, notes were taken on the participants’ approaches to the tasks and on what problems they encountered. After they had finished all the tasks, the subjects were given a questionnaire with 14 questions (7 for each tool), designed to evaluate different aspects of the tool through Likert scales. Table 6.1 shows the mean values of the answers to each question for the ScatterDice

(SD) and the Multidimensional Scatterplot (MS) tools; questions 1 to 6 have a Likert scale of 4, whereas question number 7 has a Likert scale of 5. Additionally, in conclusion of each session the participants were asked open questions to collect feedbacks and suggestions for future improvements.

Table 6.1 Tools' Comparison

Question	SD	MS
1. Is it easy to use the tool the first time?	2.6	3.5
2. Is it easy to use the tool after practicing it?	3.6	3.9
3. Was it easy to carry out the given task with the proposed tool?	3.0	3.5
4. How much difficult was the proposed task?		
task number one	2.3	1.8
task number two	2.4	2.1
task number three	2.5	2.3
5. Is the tool quick to use/examine?	2.7	3.7
6. Does the tool require a high cognitive demand to be used?	2.7	2.2
7. Give an overall score to the tool	3.8	4.5

6.6.3 Results

The tests showed that almost everyone was able to correctly complete the given task: the users were unable to provide the right answer using the multidimensional scatterplot only in two occasions, whereas with the ScatterDice it happened five times. Moreover, everyone completed the most difficult task with both tools. The multidimensional scatterplot was evaluated better than the ScatterDice in every question, even if the delta between the two visualization tools was very low for qustions two and six. The feedback provided by the user was very positive and eventually they proposed some minor changes.

6.7 Conclusions

The analysis of the state of the art allowed to define the common problems of current visualization techniques when applied to multidimensional data and a set of best practices to support the design and development of the multidimensional scatterplot adopted for the comparison. The chosen use case allowed to test the multidimensional scatterplot for different visualization tasks. Finally, a qualitative evaluation of the developed tool pointed out some benefits that such visualization paradigm could offer compared to the ScatterDice: the users are able to solve the given tasks in less time and with a lower amount of cognitive demand. Future works will include further investigations to better evaluate the efficacy of different visualization techniques to increase the dimensions displayable through the multidimensional scatterplot. Moreover, the comparison should be reinforced considering different datasets and use cases, since the distribution of values for the different data dimensions may greatly affect the visualization. Finally, it could be possible to develop additional graphical effects for the visualization of both static and dynamic data. Further test may point out which graphical effects perform better together and which ones are most suited for specific tasks in order to provide presets for the users.

References

[1] MIT CTL Visual Analytics Lab. http://val.mit.edu/. Accessed: 2017-4-2.

[2] Andrea Sanna and Federico Manuri. A survey on applications of augmented reality. *Advances in Computer Science: an International Journal*, 5(1):18–27, 2016.

[3] Fabrizio Lamberti, Federico Manuri, Andrea Sanna, Gianluca Paravati, Pietro Pezzolla, and Paolo Montuschi. Challenges, opportunities, and future trends of emerging techniques for augmented reality-based maintenance. *IEEE Transactions on Emerging Topics in Computing*, 2(4):411–421, 2014.

[4] Federico Manuri, Andrea Sanna, Fabrizio Lamberti, Gianluca Paravati, and Pietro Pezzolla. A workflow analysis for implementing ar-based maintenance procedures. In *International Conference on Augmented and Virtual Reality*, pages 185–200. Springer, 2014.

[5] Institute of Electrical and Electronics Engineers. https://www.ieee.org/index.html. Accessed: 2017-4-1.

[6] Association for Computing Machinery. https://www.acm.org/. Accessed: 2017-4-1.

[7] James H Carlisle. Evaluating the impact of office automation on top management communication. In *Proceedings of the June 7-10, 1976, national computer conference and exposition*, pages 611–616. ACM, 1976.

[8] Stuart K Card, Allen Newell, and Thomas P Moran. The psychology of human-computer interaction. 1983.

[9] Paul Milgram and Fumio Kishino. A taxonomy of mixed reality visual displays. *IEICE TRANSACTIONS on Information and Systems*, 77(12):1321–1329, 1994.

[10] Ronald T Azuma. A survey of augmented reality. *Presence: Teleoperators and virtual environments*, 6(4):355–385, 1997.

[11] VR and vice: Are we heading for mass addiction to virtual reality fantasies? https://www.wareable.com/vr/vr-and-vice-9232.

[12] Paul Milgram, Haruo Takemura, Akira Utsumi, and Fumio Kishino. Augmented reality: A class of displays on the reality-virtuality continuum. In *Photonics for industrial applications*, pages 282–292. International Society for Optics and Photonics, 1995.

[13] D Van Krevelen and R Poelman. Augmented reality: Technologies, applications, and limitations. 2007.

[14] Ivan E Sutherland. The ultimate display. *Multimedia: From Wagner to virtual reality*, 1965.

[15] Ivan E Sutherland. A head-mounted three dimensional display. In *Proceedings of the December 9-11, 1968, fall joint computer conference, part I*, pages 757–764. ACM, 1968.

[16] Thomas P Caudell and David W Mizell. Augmented reality: An application of heads-up display technology to manual manufacturing processes. In *System Sciences, 1992. Proceedings of the Twenty-Fifth Hawaii International Conference on*, volume 2, pages 659–669. IEEE, 1992.

[17] Steven Feiner, Blair Macintyre, and Dorée Seligmann. Knowledge-based augmented reality. *Communications of the ACM*, 36(7):53–62, 1993.

[18] Tobias Höllerer, Steven Feiner, Tachio Terauchi, Gus Rashid, and Drexel Hallaway. Exploring mars: developing indoor and outdoor user interfaces to a mobile augmented reality system. *Computers & Graphics*, 23(6):779–785, 1999.

[19] Wolfgang Wohlgemuth and Gunthard Triebfürst. Arvika: Augmented reality for development, production and service. In *Proceedings of DARE 2000 on Designing Augmented Reality Environments*, DARE '00, pages 151–152, New York, NY, USA, 2000. ACM.

[20] Mixed Reality Lab. https://www.nottingham.ac.uk/research/groups/mixedrealitylab/. Accessed: 2017-3-10.

[21] Ronald Azuma, Yohan Baillot, Reinhold Behringer, Steven Feiner, Simon Julier, and Blair MacIntyre. Recent advances in augmented reality. *IEEE computer graphics and applications*, 21(6):34–47, 2001.

[22] International Symposium on Mixed and Augmented Reality. http://www.ismar.net/. Accessed: 2017-3-10.

[23] Fred D Davis. Perceived usefulness, perceived ease of use, and user acceptance of information technology. *MIS quarterly*, pages 319–340, 1989.

[24] Google Glass project website. https://developers.google.com/glass/. Accessed: 2017-3-13.

[25] Microsoft Hololens. https://www.microsoft.com/microsoft-hololens/en-us. Accessed: 2017-4-1.

[26] Innovega website. http://www.emacula.io/. Accessed: 2017-3-13.

[27] Eric Foxlin et al. Motion tracking requirements and technologies. *Handbook of virtual environment technology*, 8:163–210, 2002.

[28] Hiroshi Ishii and Brygg Ullmer. Tangible bits: towards seamless interfaces between people, bits and atoms. In *Proceedings of the ACM SIGCHI Conference on Human factors in computing systems*, pages 234–241. ACM, 1997.

[29] Angela Chang and Conor O'Sullivan. Audio-haptic feedback in mobile phones. In *CHI'05 extended abstracts on Human factors in computing systems*, pages 1264–1267. ACM, 2005.

[30] Gunther Heidemann, Holger Bekel, Ingo Bax, and Helge Ritter. Interactive online learning. *Pattern Recognition and Image Analysis*, 15(1):55–58, 2005.

[31] Jack M Loomis, Reginald G Golledge, Roberta L Klatzky, Jon M Speigle, and Jerome Tietz. Personal guidance system for the visually impaired. In *Proceedings of the first annual ACM conference on Assistive technologies*, pages 85–91. ACM, 1994.

[32] Frank A Biocca and Jannick P Rolland. Virtual eyes can rearrange your body: Adaptation to visual displacement in see-through, head-mounted displays. *Presence: Teleoperators and Virtual Environments*, 7(3):262–277, 1998.

[33] Arthur Tang, Charles Owen, Frank Biocca, and Weimin Mou. Comparative effectiveness of augmented reality in object assembly. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 73–80. ACM, 2003.

[34] Akinari Takagi, Shoichi Yamazaki, Yoshihiro Saito, and Naosato Taniguchi. Development of a stereo video see-through hmd for ar systems. In *Augmented Reality, 2000.(ISAR 2000). Proceedings. IEEE and ACM International Symposium on*, pages 68–77. IEEE, 2000.

[35] Stephen R Ellis, Francois Breant, B Manges, Richard Jacoby, and Bernard D Adelstein. Factors influencing operator interaction with virtual objects viewed via head-mounted see-through displays: viewing conditions and rendering latency. In *Virtual Reality Annual International Symposium, 1997., IEEE 1997*, pages 138–145. IEEE, 1997.

[36] Kiyoshi Kiyokawa, Mark Billinghurst, Bruce Campbell, and Eric Woods. An occlusion-capable optical see-through head mount display for supporting co-located collaboration. In *Proceedings of the 2nd IEEE/ACM International Symposium on Mixed and Augmented Reality*, page 133. IEEE Computer Society, 2003.

[37] Brian T Schowengerdt, Eric J Seibel, John P Kelly, Nicholas L Silverman, and Thomas A Furness III. Binocular retinal scanning laser display with integrated focus cues for ocular accommodation. In *Electronic Imaging 2003*, pages 1–9. International Society for Optics and Photonics, 2003.

[38] Oliver Bimber and Ramesh Raskar. *Spatial augmented reality: merging real and virtual worlds*. CRC press, 2005.

[39] Ramesh Raskar, Greg Welch, and Wei-Chao Chen. Table-top spatially-augmented realty: bringing physical models to life with projected imagery. In *Augmented Reality, 1999.(IWAR'99) Proceedings. 2nd IEEE and ACM International Workshop on*, pages 64–71. IEEE, 1999.

[40] Ozan Cakmakci and Jannick Rolland. Head-worn displays: a review. *Journal of display technology*, 2(3):199–216, 2006.

[41] Wolfgang Narzt, Gustav Pomberger, Alois Ferscha, Dieter Kolb, Reiner Müller, Jan Wieghardt, Horst Hörtner, and Christopher Lindinger. Pervasive information acquisition for mobile ar-navigation systems. In *WMCSA*, volume 3, pages 13–20, 2003.

[42] Tetsuro Ogi, Toshio Yamada, Koji Yamamoto, and Michitaka Hirose. Invisible interface for the immersive virtual world. In *Immersive Projection Technology and Virtual Environments 2001*, pages 237–246. Springer, 2001.

[43] H Steinhaus. Sur la localisation au moyen des rayons x. *Comptes Rendus de L'Academie des Science*, 206:1473–5, 1938.

[44] Tobias Sielhorst, Marco Feuerstein, and Nassir Navab. Advanced medical displays: A literature review of augmented reality. *Journal of Display Technology*, 4(4):451–467, 2008.

[45] Steven Henderson and Steven Feiner. Exploring the benefits of augmented reality documentation for maintenance and repair. *IEEE transactions on visualization and computer graphics*, 17(10):1355–1368, 2011.

[46] Bruce Thomas, Benjamin Close, John Donoghue, John Squires, Phillip De Bondi, Michael Morris, and Wayne Piekarski. Arquake: An outdoor/indoor augmented reality first person application. In *Wearable computers, the fourth international symposium on*, pages 139–146. IEEE, 2000.

[47] Rick Cavallaro, Maria Hybinette, Marvin White, and Tucker Balch. Augmenting live broadcast sports with 3d tracking information. *IEEE MultiMedia*, 18(4):38–47, 2011.

[48] Mark Billinghurst and Hirokazu Kato. Collaborative augmented reality. *Communications of the ACM*, 45(7):64–70, 2002.

[49] Zsolt Szalavári, Dieter Schmalstieg, Anton Fuhrmann, and Michael Gervautz. "studierstube": An environment for collaboration in augmented reality. *Virtual Reality*, 3(1):37–48, 1998.

[50] Kevin Hannam, Gareth Butler, and Cody Morris Paris. Developments and key issues in tourism mobilities. *Annals of Tourism Research*, 44:171–185, 2014.

[51] Chris D Kounavis, Anna E Kasimati, and Efpraxia D Zamani. Enhancing the tourism experience through mobile augmented reality: Challenges and prospects. *International Journal of Engineering Business Management*, 4:10, 2012.

[52] Kiyoshi Kiyokawa, Makoto Niimi, Tsuyoshi Ebina, and Hiroyuki Ohno. mr^2 (mr square): a mixed-reality meeting room. In *Augmented Reality, 2001. Proceedings. IEEE and ACM International Symposium on*, pages 169–170. IEEE, 2001.

[53] Wolfgang Broll, Irma Lindt, Jan Ohlenburg, Michael Wittkämper, Chunrong Yuan, Thomas Novotny, Chiron Mottram, A Fatah gen Schieck, and A Strothman. Arthur: A collaborative augmented environment for architectural design and urban planning. 2004.

[54] Vassilios Vlahakis, M Ioannidis, John Karigiannis, Manolis Tsotros, Michael Gounaris, Didier Stricker, Tim Gleue, Patrick Daehne, and Luís Almeida. Archeoguide: an augmented reality guide for archaeological sites. *IEEE Computer Graphics and Applications*, 22(5):52–60, 2002.

[55] Kajos Attila and Bányai Edit. Beyond reality: The possibilities of augmented reality in cultural and heritage tourism. In *2nd International Tourism and Sport Management Conference, Debrecen*, pages 5–6, 2012.

[56] S Yuen, Gallayanee Yaoyuneyong, and Erik Johnson. Augmented reality: An overview and five directions for ar in education. *Journal of Educational Technology Development and Exchange*, 4(1):119–140, 2011.

[57] Hsin-Kai Wu, Silvia Wen-Yu Lee, Hsin-Yi Chang, and Jyh-Chong Liang. Current status, opportunities and challenges of augmented reality in education. *Computers & Education*, 62:41–49, 2013.

[58] Mark A Livingston, Lawrence J Rosenblum, Simon J Julier, Dennis Brown, Yohan Baillot, II Swan, Joseph L Gabbard, Deborah Hix, et al. An augmented reality system for military operations in urban terrain. Technical report, DTIC Document, 2002.

[59] Zhiwei Zhu, Vlad Branzoi, Mikhail Sizintsev, Nicholas Vitovitch, Taragay Oskiper, Ryan Villamil, Ali Chaudhry, Supun Samarasekera, and Rakesh Kumar. Ar-weapon: live augmented reality based first-person shooting system. In *Applications of Computer Vision (WACV), 2015 IEEE Winter Conference on*, pages 618–625. IEEE, 2015.

[60] Fabrizio Lamberti, Federico Manuri, Gianluca Paravati, Giovanni Piumatti, and Andrea Sanna. Using semantics to automatically generate speech interfaces for wearable virtual and augmented reality applications. *IEEE Transactions on Human-Machine Systems*, 2016.

[61] Andrea Sanna, Federico Manuri, Giovanni Piumatti, Gianluca Paravati, Fabrizio Lamberti, and Pietro Pezzolla. A flexible ar-based training system for industrial maintenance. In *International Conference on Augmented and Virtual Reality*, pages 314–331. Springer, 2015.

[62] Federico Manuri, Andrea Sanna, Fabrizio Lamberti, and Gianluca Paravati. Vocal one switch (vos) selection interfaces for virtual and augmented reality hands-free tasks.

[63] GP Sullivan, R Pugh, AP Melendez, and WD Hunt. Operations & maintenance best practices. *A guide to achieving operational efficiency, Release*, 2, 2004.

[64] T Claypole and N Wells. Productivity maintenance in the uk printing industry: A cross-industry survey. *Vision in Print*, 2005.

[65] U.S.AE]xecutive Order 13514, howpublished=https://www.fedcenter.gov/programs/eo13514/, note=Accessed: 2017-4-1.

[66] GP Sullivan, R Pugh, AP Melendez, and WD Hunt. Operations & maintenance best practices – a guide to achieving operational efficiency (release 3.0), 2010.

[67] Amik Garg and SG Deshmukh. Maintenance management: literature review and directions. *Journal of Quality in Maintenance Engineering*, 12(3):205–238, 2006.

[68] Anis Chelbi and Daoud Ait-Kadi. Analysis of a production/inventory system with randomly failing production unit submitted to regular preventive maintenance. *European journal of operational research*, 156(3):712–718, 2004.

[69] Muh-Guey Juang and Gary Anderson. A bayesian method on adaptive preventive maintenance problem. *European Journal of Operational Research*, 155(2):455–473, 2004.

[70] Usgame Sandoval, CA Pedraza Ramirez, JE Quiroga Méndez, et al. Acoustic emission-based early fault detection in tapered roller bearings. *Ingeniería e Investigación*, 33(3):5–10, 2013.

[71] Nassir Navab. Developing killer apps for industrial augmented reality. *IEEE Computer Graphics and applications*, 24(3):16–20, 2004.

[72] What that Car Really Costs to Own. http://www.consumerreports.org/cro/2012/12/what-that-car-really-costs-to-own/index.htm. Accessed: 2017-3-23.

[73] Jung-Min Lee, Kyung-Ho Lee, Dae-Seok Kim, Gwang Lee, et al. An enhanced smart maintenance of piping system for the offshore plants. In *The Twenty-second International Offshore and Polar Engineering Conference*. International Society of Offshore and Polar Engineers, 2012.

[74] SG Lee, Y-S Ma, GL Thimm, and J Verstraeten. Product lifecycle management in aviation maintenance, repair and overhaul. *Computers in industry*, 59(2):296–303, 2008.

[75] Joseph J Fuller. Ietms: From research to reality. *CALS Expo'94 International Proceedings*, 1994.

[76] Steven J Henderson and Steven K Feiner. Augmented reality for maintenance and repair (armar). Technical report, DTIC Document, 2007.

[77] James Ott. Maintenance executives seek greater efficiency. *Aviation Week & Space Technology*, 142(20):43–44, 1995.

[78] Douglas M Towne. Cognitive workload in fault diagnosis. Technical report, DTIC Document, 1985.

[79] Robert D Rogers and Stephen Monsell. Costs of a predictible switch between simple cognitive tasks. *Journal of experimental psychology: General*, 124(2):207, 1995.

[80] J A Waterworth and year=1997 publisher=Elsevier Chignell, M H] book=Handbook of Human-Computer Interaction. Multimedia Interaction.

[81] Elizabeth S Veinott, Barbara G Kanki, and Michael G Shafto. Identifying human factors issues in aircraft maintenance operations. 1995.

[82] A Fisher. Danger zone, 1997.

[83] G Terenzi and G Basile. Smart maintenance: An augmented reality platform for training and fields operations in the manufacturing industry. armedia augmented reality blog (2014). http://arblog.inglobetechnologies.com/?p=1370. Accessed: 2017-3-22.

[84] Johannes Tumler, Fabian Doil, Rudiger Mecke, Georg Paul, Michael Schenk, Eberhard A Pfister, Anke Huckauf, Irina Bockelmann, and Anja Roggentin. Mobile augmented reality in industrial applications: Approaches for solution of user-related issues. In *Mixed and Augmented Reality, 2008. ISMAR 2008. 7th IEEE/ACM International Symposium on*, pages 87–90. IEEE, 2008.

[85] SK Ong, ML Yuan, and AYC Nee. Augmented reality applications in manufacturing: a survey. *International journal of production research*, 46(10):2707–2742, 2008.

[86] AYC Nee, SK Ong, G Chryssolouris, and D Mourtzis. Augmented reality applications in design and manufacturing. *CIRP Annals-manufacturing technology*, 61(2):657–679, 2012.

[87] *Fundamentals of wearable computers and augmented reality*. CRC Press, 2015.

[88] Sylvia Irawati, Scott Green, Mark Billinghurst, Andreas Duenser, and Hee-dong Ko. Move the couch where?: Developing an augmented reality multimodal interface. In *Proceedings of the 5th IEEE and ACM International Symposium on Mixed and Augmented Reality*, pages 183–186. IEEE Computer Society, 2006.

[89] Matthias Kölsch, Ryan Bane, Tobias Höllerer, and Matthew Turk. Touching the visualized invisible: Wearable ar with a multimodal interface. *IEEE Computer Graphics and Applications*, 26(3):62–71, 2006.

[90] Michael Kleiber, Thomas Alexander, Carsten Winkelholz, and Christopher M Schlick. User-centered design and evaluation of an integrated ar-vr system for tele-maintenance. In *Systems, Man, and Cybernetics (SMC), 2012 IEEE International Conference on*, pages 1443–1448. IEEE, 2012.

[91] Wolfgang Friedrich, D Jahn, and L Schmidt. Arvika-augmented reality for development, production and service. In *ISMAR*, volume 2002, pages 3–4. Citeseer, 2002.

[92] Bernd Schwald, Jean Figue, Eric Chauvineau, Franck Vu-Hong, Anne Robert, Marco Arbolino, et al. Starmate: Using augmented reality technology for computer guided maintenance of complex mechanical elements. *E-work and ECommerce*, 1:196–202, 2001.

[93] Panu Harmo, Aarne Halme, Petri Virekoski, Matias Halinen, and Hannu Pitkänen. Etälä-virtual reality assisted telepresence system for remote maintenance. In *Proceedings of the 1st IFAC conference on Mechatronic Systems, Darmstadt, Germany*, pages 1075–1080, 2000.

[94] Carlos Rodriguez Sanchez, Francisco Javier Gonzalez Fernandez, Laura Carmen Simon Vena, Jose Carpio, and Manuel Castro. Industrial telemaintenance: Remote management experience from subway to industrial electronics. *IEEE Transactions on Industrial Electronics*, 58(3):1044–1051, 2011.

[95] Ning Wang and Yue Qi. Virtual assembly, maintenance and training system based on the virtual-real fusion technology. In *Green Computing and Communications (GreenCom), 2013 IEEE and Internet of Things (iThings/CPSCom), IEEE International Conference on and IEEE Cyber, Physical and Social Computing*, pages 1949–1952. IEEE, 2013.

[96] Steven J Henderson and Steven Feiner. Evaluating the benefits of augmented reality for task localization in maintenance of an armored personnel carrier turret. In *Mixed and Augmented Reality, 2009. ISMAR 2009. 8th IEEE International Symposium on*, pages 135–144. IEEE, 2009.

[97] Steven Henderson and Steven Feiner. Opportunistic tangible user interfaces for augmented reality. *IEEE Transactions on Visualization and Computer Graphics*, 16(1):4–16, 2010.

[98] Changzhi Ke, Bo Kang, Dongyi Chen, and Xinyu Li. An augmented reality-based application for equipment maintenance. In *International Conference on Affective Computing and Intelligent Interaction*, pages 836–841. Springer, 2005.

[99] The ARTESAS Project. http://www.mip.informatik.uni-kiel.de/tiki-index.php?page=Artesas. Accessed: 2017-3-22.

[100] Bernd Schwald and Blandine De Laval. An augmented reality system for training and assistance to maintenance in the industrial context. 2003.

[101] Alex Olwal, Jonny Gustafsson, and Christoffer Lindfors. Spatial augmented reality on industrial cnc-machines. In *Electronic Imaging 2008*, pages 680409–680409. International Society for Optics and Photonics, 2008.

[102] Gandjar Kiswanto and Dedy Ariansyah. Development of augmented reality (ar) for machining simulation of 3-axis cnc milling. In *Advanced Computer Science and Information Systems (ICACSIS), 2013 International Conference on*, pages 143–148. IEEE, 2013.

[103] Juha Sääski, Tapio Salonen, Marja Liinasuo, Jarkko Pakkanen, Mikko Vanhatalo, and Asko Riitahuhta. Augmented reality efficiency in manufacturing industry: a case study. In *DS 50: Proceedings of NordDesign 2008 Conference, Tallinn, Estonia, 21.-23.08. 2008*, 2008.

[104] Tom Haritos and Nickolas D Macchiarella. A mobile application of augmented reality for aerospace maintenance training. In *Digital Avionics Systems Conference, 2005. DASC 2005. The 24th*, volume 1, pages 5–B. IEEE, 2005.

[105] Paula Savioja, Paula Järvinen, Tommi Karhela, Pekka Siltanen, and Charles Woodward. Developing a mobile, service-based augmented reality tool for modern maintenance work. In *International Conference on Virtual Reality*, pages 554–563. Springer, 2007.

[106] Katharina Pentenrieder, Christian Bade, Fabian Doil, and Peter Meier. Augmented reality-based factory planning-an application tailored to industrial needs. In *Mixed and Augmented Reality, 2007. ISMAR 2007. 6th IEEE and ACM International Symposium on*, pages 31–42. IEEE, 2007.

[107] Sanghoon Lee and Ömer Akin. Augmented reality-based computational fieldwork support for equipment operations and maintenance. *Automation in Construction*, 20(4):338–352, 2011.

[108] D. Wagnerč. *andheld augmented reality*. PhD thesis, Graz University of Technology, Institute of Computer Graphics and Vision, 2007.

[109] Dieter Schmalstieg and Daniel Wagner. Experiences with handheld augmented reality. In *Mixed and Augmented Reality, 2007. ISMAR 2007. 6th IEEE and ACM International Symposium on*, pages 3–18. IEEE, 2007.

[110] Advancing asset knowledge through the use of augmented reality technologies. http://www.fiatech.org/webinars/2013-webinars. Accessed: 2017-3-22.

[111] Svenja Kahn, Manuel Olbrich, Timo Engelke, Jens Keil, Patrick Riess, Sabine Webel, Holger Graf, Ulrich Bockholt, and Guillaume Picinbono. Beyond 3d" as-built" information using mobile ar enhancing the building lifecycle management. In *Cyberworlds (CW), 2012 International Conference on*, pages 29–36. IEEE, 2012.

[112] Sabine Webel, Uli Bockholt, Timo Engelke, Matteo Peveri, Manuel Olbrich, and Carsten Preusche. Augmented reality training for assembly and maintenance skills. In *BIO Web of Conferences*, volume 1, page 00097. EDP Sciences, 2011.

[113] Sabine Webel, Uli Bockholt, Timo Engelke, Nirit Gavish, Manuel Olbrich, and Carsten Preusche. An augmented reality training platform for assembly and maintenance skills. *Robotics and Autonomous Systems*, 61(4):398–403, 2013.

[114] Luis Eduardo Garza, Gabriel Pantoja, Pablo Ramírez, Hector Ramírez, Nestor Rodríguez, Eduardo González, Raúl Quintal, and Juan A Pérez. Augmented reality application for the maintenance of a flapper valve of a fuller-kynion type m pump. *Procedia Computer Science*, 25:154–160, 2013.

[115] Yu-Doo Kim and Il-Young Moon. E-training content delivery networking system for augmented reality car maintenance training application. *International Journal of Multimedia and Ubiquitous Engineering*, 8(2):69–80, 2013.

[116] Lukas Van Campenhout, Joep Frens, Kees Overbeeke, Achiel Standaert, and Herbert Peremans. Physical interaction in a dematerialized world. *International Journal of Design*, 7(1), 2013.

[117] Augmented reality app from Audi sidesteps A3 owner manual. http://www.augmentedrealitytrends.com/ar-app/augmented-reality-app-from-audi-sidesteps-a3-owner-manual.html. Accessed: 2017-3-22.

[118] Tapio Salonen and Juha Sääski. Dynamic and visual assembly instruction for configurable products using augmented reality techniques. In *Advanced Design and Manufacture to Gain a Competitive Edge*, pages 23–32. Springer, 2008.

[119] J Servan, F Mas, JL Menéndez, and J Ríos. Assembly work instruction deployment using augmented reality. In *Key Engineering Materials*, volume 502, pages 25–30. Trans Tech Publ, 2012.

[120] Lei Hou, Xiangyu Wang, Leonhard Bernold, and Peter ED Love. Using animated augmented reality to cognitively guide assembly. *Journal of Computing in Civil Engineering*, 27(5):439–451, 2013.

[121] L Stinson. So smart: New Ikea app places virtual furniture in your home. https://www.wired.com/2013/08/a-new-ikea-app-lets-you-place-3d-furniture-in-your-home/. Accessed: 2017-3-22.

[122] insidear. http://www.metaio.eu/fileadmin/upload/documents/magazine/insidear_mag-005-2013.pdf. Accessed: 2017-5-12.

[123] Johannes Christian, Horst Krieger, Andreas Holzinger, and Reinhold Behringer. Virtual and mixed reality interfaces for e-training: examples of applications in light aircraft maintenance. In *International Conference on Universal Access in Human-Computer Interaction*, pages 520–529. Springer, 2007.

[124] Kamyar Abhari, John SH Baxter, Elvis CS Chen, Ali R Khan, Terry M Peters, Sandrine de Ribaupierre, and Roy Eagleson. Training for planning tumour resection: augmented reality and human factors. *IEEE Transactions on Biomedical Engineering*, 62(6):1466–1477, 2015.

[125] David Liu, Simon A Jenkins, Penelope M Sanderson, Marcus O Watson, Terrence Leane, Amanda Kruys, and W John Russell. Monitoring with head-mounted displays: performance and safety in a full-scale simulator and part-task trainer. *Anesthesia & Analgesia*, 109(4):1135–1146, 2009.

[126] Fotis Liarokapis, Panos Petridis, Paul F Lister, Martin White, et al. Multimedia augmented reality interface for e-learning (marie). *World Transactions on Engineering and Technology Education*, 1(2):173–176, 2002.

[127] Fotis Liarokapis, Nikolaos Mourkoussis, Martin White, Joe Darcy, Maria Sifniotis, Panos Petridis, Anirban Basu, and Paul F Lister. Web3d and augmented reality to support engineering education. *World Transactions on Engineering and Technology Education*, 3(1):11–14, 2004.

[128] Mark Billinghurst and Anders Henrysson. Mobile architectural augmented reality. In *Mixed reality in architecture, design and construction*, pages 93–104. Springer, 2009.

[129] Jon Cook, Simon Gibson, Toby Howard, and Roger Hubbold. Real-time photo-realistic augmented reality for interior design. In *ACM SIGGRAPH 2003 Sketches & Applications*, pages 1–1. ACM, 2003.

[130] Mannu Amrit, Himanshu Bansal, and Pradeep Yammiyavar. Studies in application of augmented reality in e-learning courses. In *ICoRD'15–Research into Design Across Boundaries Volume 2*, pages 375–384. Springer, 2015.

[131] Yu-Chien Chen. A study of comparing the use of augmented reality and physical models in chemistry education. In *Proceedings of the 2006 ACM international conference on Virtual reality continuum and its applications*, pages 369–372. ACM, 2006.

[132] Hannes Kaufmann. Construct3d: an augmented reality application for mathematics and geometry education. In *Proceedings of the tenth ACM international conference on Multimedia*, pages 656–657. ACM, 2002.

[133] Hannes Kaufmann and Dieter Schmalstieg. Mathematics and geometry education with collaborative augmented reality. *Computers & Graphics*, 27(3):339–345, 2003.

[134] Changhun Chae and Kwanghee Ko. Introduction of physics simulation in augmented reality. In *Ubiquitous Virtual Reality, 2008. ISUVR 2008. International Symposium on*, pages 37–40. IEEE, 2008.

[135] Hannes Kaufmann and Bernd Meyer. *Simulating educational physical experiments in augmented reality*. ACM, 2008.

[136] H Kim, O Yoon, J Han, and K Nam. Distributed cognition-applied smart learning environment: Design and implementation. In *EDULEARN14 Proceedings*, pages 381–381. IATED, 2014.

[137] Brett E Shelton and Nicholas R Hedley. Using augmented reality for teaching earth-sun relationships to undergraduate geography students. In *Augmented Reality Toolkit, The First IEEE International Workshop*, pages 8–pp. IEEE, 2002.

[138] Ioannis Deliyiannis and Georgios Papaioannou. Augmented reality for archaeological environments on mobile devices: a novel open framework. *Mediterranean Archaeology and Archaeometry*, 14(4):1–10, 2014.

[139] Chung Hsien Tsai and Jiung Yao Huang. A mobile augmented reality based scaffolding platform for outdoor fieldtrip learning. In *Advanced Applied Informatics (IIAIAAI), 2014 IIAI 3rd International Conference on*, pages 307–312. IEEE, 2014.

[140] Ángela Di Serio, María Blanca Ibáñez, and Carlos Delgado Kloos. Impact of an augmented reality system on students' motivation for a visual art course. *Computers & Education*, 68:586–596, 2013.

[141] Leticia Gomes, Valeria Farinazzo Martins, Diego Colombo Dias, and Marcelo de Paiva Guimarães. Music-ar: augmented reality in teaching the concept of sound loudness to children in pre-school. In *Virtual and Augmented Reality (SVR), 2014 XVI Symposium on*, pages 114–117. IEEE, 2014.

[142] Dragoş Datcu, Marina Cidota, Stephan Lukosch, David Martinez Oliveira, and Mikael Wolff. Virtual co-location to support remote assistance for inflight maintenance in ground training for space missions. In *Proceedings of the 15th International Conference on Computer Systems and Technologies*, pages 134–141. ACM, 2014.

[143] Francesca De Crescenzio, Massimiliano Fantini, Franco Persiani, Luigi Di Stefano, Pietro Azzari, and Samuele Salti. Augmented reality for aircraft maintenance training and operations support. *IEEE Computer Graphics and Applications*, 31(1):96–101, 2011.

[144] Darko Stanimirovic, Nina Damasky, Sabine Webel, Dirk Koriath, Andrea Spillner, and Daniel Kurz. [poster] a mobile augmented reality system to assist auto mechanics. In *Mixed and Augmented Reality (ISMAR), 2014 IEEE International Symposium on*, pages 305–306. IEEE, 2014.

[145] Junfeng Wang, Yaqing Feng, Cheng Zeng, and Shiqi Li. An augmented reality based system for remote collaborative maintenance instruction of complex products. In *Automation Science and Engineering (CASE), 2014 IEEE International Conference on*, pages 309–314. IEEE, 2014.

[146] Stuart Goose, Sandra Sudarsky, Xiang Zhang, and Nassir Navab. Speech-enabled augmented reality supporting mobile industrial maintenance. *IEEE Pervasive Computing*, 2(1):65–70, 2003.

[147] Gabriel Pantoja, Luis Eduardo Garza, and Eduardo Gonzalez Mendivil. Augmented reality in pneumatic conveying system: fuller pump dry material line charger. In *Information Systems and Technologies (CISTI), 2014 9th Iberian Conference on*, pages 1–5. IEEE, 2014.

[148] Mark Billinghurst, Hirokazu Kato, and Ivan Poupyrev. The magicbook-moving seamlessly between reality and virtuality. *IEEE Computer Graphics and applications*, 21(3):6–8, 2001.

[149] Galina Ivanova, Yuksel Aliev, and Aleksandar Ivanov. Augmented reality textbook for future blended education. In *Proceedings of the International Conference on E-Learning*, volume 14, pages 130–136, 2014.

[150] Adrian David Cheok, Siew Wan Fong, Kok Hwee Goh, Xubo Yang, Wei Liu, Farzam Farzbiz, and Yu Li. Human pacman: A mobile entertainment system with ubiquitous computing and tangible interaction over a wide outdoor area. In *International Conference on Mobile Human-Computer Interaction*, pages 209–223. Springer, 2003.

[151] T Ohshima, K Satoh, H Yamamoto, and H Tamura. Ar2 hockey system: A collaborative mixed reality system. *Trans. VRSJ*, 3(2):55–60, 1998.

[152] Amir H Behzadan and Vineet R Kamat. Enabling discovery-based learning in construction using telepresent augmented reality. *Automation in Construction*, 33:3–10, 2013.

[153] Ji-Sun Kim, Denis Gracanin, Kresimir Matkovic, and Francis Quek. iphone/ipod touch as input devices for navigation in immersive virtual environments. In *Virtual Reality Conference, 2009. VR 2009. IEEE*, pages 261–262. IEEE, 2009.

[154] Sunao Hashimoto, Akihiko Ishida, Masahiko Inami, and Takeo Igarashi. Touchme: an augmented reality interface for remote robot control. *Journal of Robotics and Mechatronics*, 25(3):529–537, 2013.

[155] Federico Manuri and Giovanni Piumatti. A preliminary study of a hybrid user interface for augmented reality applications. In *Intelligent Technologies for Interactive Entertainment (INTETAIN), 2015 7th International Conference on*, pages 37–41. IEEE, 2015.

[156] Pranav Mistry and Pattie Maes. Sixthsense: a wearable gestural interface. In *ACM SIGGRAPH ASIA 2009 Sketches*, page 11. ACM, 2009.

[157] Pranav Mistry, Pattie Maes, and Liyan Chang. Wuw-wear ur world: a wearable gestural interface. In *CHI'09 extended abstracts on Human factors in computing systems*, pages 4111–4116. ACM, 2009.

[158] Holger Regenbrecht, Jonny Collins, and Simon Hoermann. A leap-supported, hybrid ar interface approach. In *Proceedings of the 25th Australian Computer-Human Interaction Conference: Augmentation, Application, Innovation, Collaboration*, pages 281–284. ACM, 2013.

[159] Andrea Sanna, Fabrizio Lamberti, Federica Bazzano, and Luigi Maggio. Developing touch-less interfaces to interact with 3d contents in public exhibitions. In *International Conference on Augmented Reality, Virtual Reality and Computer Graphics*, pages 293–303. Springer, 2016.

[160] Peter Dam, Priscilla Braz, and Alberto Raposo. A study of navigation and selection techniques in virtual environments using microsoft kinect®. In *International Conference on Virtual, Augmented and Mixed Reality*, pages 139–148. Springer, 2013.

[161] Gun A Lee, Claudia Nelles, Mark Billinghurst, and Gerard Jounghyun Kim. Immersive authoring of tangible augmented reality applications. In *Proceedings of the 3rd IEEE/ACM international Symposium on Mixed and Augmented Reality*, pages 172–181. IEEE Computer Society, 2004.

[162] Mark Billinghurst, Hirokazu Kato, and Ivan Poupyrev. Tangible augmented reality. *ACM SIGGRAPH ASIA*, 7, 2008.

[163] Hyung Min Park, Seok Han Lee, and Jong Soo Choi. Wearable augmented reality system using gaze interaction. In *Proceedings of the 7th IEEE/ACM International Symposium on Mixed and Augmented Reality*, pages 175–176. IEEE Computer Society, 2008.

[164] Toshikazu Ohshima, Hiroyuki Yamamoto, and Hideyuki Tamura. Gaze-directed adaptive rendering for interacting with virtual space. In *Virtual reality annual international symposium, 1996., Proceedings of the IEEE 1996*, pages 103–110. IEEE, 1996.

[165] Phil J Bartie and William A Mackaness. Development of a speech-based augmented reality system to support exploration of cityscape. *Transactions in GIS*, 10(1):63–86, 2006.

[166] Anatole Lécuyer, Fabien Lotte, Richard B Reilly, Robert Leeb, Michitaka Hirose, and Mel Slater. Brain-computer interfaces, virtual reality, and videogames. *Computer*, 41(10), 2008.

[167] Karla Felix Navarro. Wearable, wireless brain computer interfaces in augmented reality environments. In *Information Technology: Coding and Computing, 2004. Proceedings. ITCC 2004. International Conference on*, volume 2, pages 643–647. IEEE, 2004.

[168] Scott R Klemmer, Anoop K Sinha, Jack Chen, James A Landay, Nadeem Aboobaker, and Annie Wang. Suede: a wizard of oz prototyping tool for speech user interfaces. In *Proceedings of the 13th annual ACM symposium on User interface software and technology*, pages 1–10. ACM, 2000.

[169] J Ferreiros, JM Pardo, R De Córdoba, Javier Macias-Guarasa, JM Montero, F Fernández, Valentin Sama, G González, et al. A speech interface for air traffic control terminals. *Aerospace Science and Technology*, 21(1):7–15, 2012.

[170] Daniel Sonntag, Sonja Zillner, Christian Schulz, Markus Weber, and Takumi Toyama. Towards medical cyber-physical systems: Multimodal augmented reality for doctors and knowledge discovery about patients. In *International Conference of Design, User Experience, and Usability*, pages 401–410. Springer, 2013.

[171] JP Woodard and EJ Cupples. Selected military applications of automatic speech recognition technology. *IEEE Communications Magazine*, 21:35–41, 1983.

[172] Jerome White and Mayuri Duggirala. Speech-interface prompt design: lessons from the field. In *Proceedings of the Seventh International Conference on Information and Communication Technologies and Development*, page 66. ACM, 2015.

[173] Nicole Yankelovich. How do users know what to say? *interactions*, 3(6):32–43, 1996.

[174] D Newman. Speech interfaces that require less human memory. *Speech Technology*, 2000.

[175] Min Yin and Shumin Zhai. The benefits of augmenting telephone voice menu navigation with visual browsing and search. In *Proceedings of the SIGCHI conference on Human Factors in computing systems*, pages 319–328. ACM, 2006.

[176] Stephan Gamm and R Haeb-Umbach. User interface design of voice controlled consumer electronics. *Philips Journal of Research*, 49(4):439–454, 1995.

[177] Catalina Danis, Liam Comerford, Eric Janke, Ken Davies, Jackie De Vries, and Alex Bertrand. Storywriter: A speech oriented editor. In *Conference companion on Human factors in computing systems*, pages 277–278. ACM, 1994.

[178] Michael Payne, Rohan Coelho, and Maher Hawash. Color as a visual cue in speech-enabled applications, September 25 2001. US Patent App. 09/965,230.

[179] Stefanie Shriver and Roni Rosenfeld. Keywords for a universal speech interface. In *CHI'02 Extended Abstracts on Human Factors in Computing Systems*, pages 726–727. ACM, 2002.

[180] Metaio. http://www.metaio.eu/. Accessed: 2017-4-2.

[181] Vuforia. https://www.vuforia.com/. Accessed: 2017-4-2.

[182] ARtoolkit. hhttps://artoolkit.org/. Accessed: 2017-4-2.

[183] P Ducher. Interaction with augmented reality. *Advances in Embedded Interactive Systems*, 2:23–29, 2014.

[184] Jacek Jankowski and Martin Hachet. A survey of interaction techniques for interactive 3d environments. In *Eurographics 2013-STAR*, 2013.

[185] Andreas D$\ddot{\nu}$nser, Raphaël Grasset, Hartmut Seichter, and Mark Billinghurst. Applying hci principles to ar systems design. *HIT Lab NZ, University of Canterbury, New Zealand*, 2004.

[186] R Teather and W Stuerzlinger. The challenge of 3d interaction: Guidelines for intuitive 3d manipulation techniques, presentation at interacting with immersive worlds, 2007.

[187] Lucinda Kerawalla, Rosemary Luckin, Simon Seljeflot, and Adrian Woolard. "making it real": exploring the potential of augmented reality for teaching primary school science. *Virtual Reality*, 10(3-4):163–174, 2006.

[188] FIDIA. http://www.fidia.it/index_en.htm#. Accessed: 2017-4-1.

[189] WIRES. http://www.cofiplast.it/en. Accessed: 2017-4-1.

[190] International Institute of Business Analysis. *A Guide to the Business Analysis Body of Knowledge (BABOK Guide), Version 2.0*. International Institute of Business Analysis, 2009.

[191] Dai Clegg and Richard Barker. *Case method fast-track: a RAD approach*. Addison-Wesley Longman Publishing Co., Inc., 1994.

[192] Rhinocers. https://www.rhino3d.com/. Accessed: 2017-4-1.

[193] Autodesk 3D Studio Max. http://www.autodesk.com/products/3ds-max/overview. Accessed: 2017-4-1.

[194] Autodesk Inventor. http://www.autodesk.it/products/inventor/overview. Accessed: 2017-4-1.

[195] Blender. https://www.blender.org/. Accessed: 2017-4-1.

[196] Yu Zhong, TV Raman, Casey Burkhardt, Fadi Biadsy, and Jeffrey P Bigham. Justspeak: enabling universal voice control on android. In *Proceedings of the 11th Web for All Conference*, page 36. ACM, 2014.

[197] Jeffrey P Bigham, Craig M Prince, and Richard E Ladner. Webanywhere: enabling a screen reading interface for the web on any computer. In *Proceedings of the 17th international conference on World Wide Web*, pages 1159–1160. ACM, 2008.

[198] Lu Xudong and Wan Jiancheng. User interface design model. In *Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing, 2007. SNPD 2007. Eighth ACIS International Conference on*, volume 3, pages 538–543. IEEE, 2007.

[199] Takanori Mori, Takako Nonaka, and Tomohiro Hase. Automatic gui generation on av remote control using genetic algorithm. In *Consumer Electronics (ISCE), 2010 IEEE 14th International Symposium on*, pages 1–3. IEEE, 2010.

[200] Andrew Sears. Layout appropriateness: A metric for evaluating user interface widget layout. *IEEE Transactions on Software Engineering*, 19(7):707–719, 1993.

[201] Krzysztof Z Gajos, Jacob O Wobbrock, and Daniel S Weld. Automatically generating user interfaces adapted to users' motor and vision capabilities. In *Proceedings of the 20th annual ACM symposium on User interface software and technology*, pages 231–240. ACM, 2007.

[202] Fabrizio Lamberti and Andrea Sanna. Extensible guis for remote application control on mobile devices. *IEEE computer graphics and applications*, 28(4), 2008.

[203] Steve Caplin. *Icon design: Graphic icons in computer interface design*. Watson-Guptill Publications, Inc., 2001.

[204] A Donald and Stephen W Norman Draper. User centered system design: New perspectives on human-computer interaction. 1986.

[205] Sven Blankenberger and Klaus Hahn. Effects of icon design on human-computer interaction. *International Journal of Man-Machine Studies*, 35(3):363–377, 1991.

[206] Ben Shneiderman. *Designing the user interface: strategies for effective human-computer interaction*. Pearson Education India, 2010.

[207] William C Howell and Alfred H Fuchs. Population stereotypy in code design. *Organizational Behavior and Human Performance*, 3(3):310–339, 1968.

[208] Elizabeth A Heard. Symbol study-1972. Technical report, SAE Technical Paper, 1974.

[209] D Paul T Piamonte, John DA Abeysekera, and Kjell Ohlsson. Understanding small graphical symbols: a cross-cultural study. *International Journal of Industrial Ergonomics*, 27(6):399–404, 2001.

[210] Makoto Morioka, Masato Ikemori, Kazuo Hattori, and Hiroshi Suzuki. Automatic icon generation system, November 22 1994. US Patent 5,367,626.

[211] Eamonn Keogh, Li Wei, Xiaopeng Xi, Stefano Lonardi, Jin Shieh, and Scott Sirowy. Intelligent icons: Integrating lite-weight data mining and visualization into gui operating systems. In *Data Mining, 2006. ICDM'06. Sixth International Conference on*, pages 912–916. IEEE, 2006.

[212] Mizuho Oda and Takayuki Itoh. Mist: a music icon selection technique using neural network. *NICOGRAPH International*, 2007, 2007.

[213] Vidya Setlur and Jock D Mackinlay. Automatic generation of semantic icon encodings for visualizations. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 541–550. ACM, 2014.

[214] Auticons, by Heydon Pickering. shttp://heydonworks.com/auticons-icon-font/. Accessed: 2017-4-1.

[215] State Chart XML (SCXML): State machine notation for control abstraction. http://www.w3.org/TR/scxml/. Accessed: 2017-4-1.

[216] Clear Icons. http://appzgear.com/products/clear-icons.htm. Accessed: 2017-4-1.

[217] Emanuele Pianta, Luisa Bentivogli, and Christian Girardi. Developing an aligned multilingual database. In *Proc. 1st Int'l Conference on Global WordNet*, 2002.

[218] George A Miller. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41, 1995.

[219] Juan J Durillo and Antonio J Nebro. jmetal: A java framework for multi-objective optimization. *Advances in Engineering Software*, 42(10):760–771, 2011.

[220] Jennifer Angelo. Comparison of three computer scanning modes as an interface method for persons with cerebral palsy. *American Journal of Occupational Therapy*, 46(3):217–222, 1992.

[221] Tamara Broderick and David JC MacKay. Fast and flexible selection with a single switch. *PloS one*, 4(10):e7481, 2009.

[222] Sebastián Aced López, Fulvio Corno, and Luigi De Russis. Playable one-switch video games for children with severe motor disabilities based on gnomon. In *Intelligent Technologies for Interactive Entertainment (INTETAIN), 2015 7th International Conference on*, pages 176–185. IEEE, 2015.

[223] Richard C Simpson and Heidi H Koester. Adaptive one-switch row-column scanning. *IEEE Transactions on Rehabilitation Engineering*, 7(4):464–473, 1999.

[224] Yujin Wakita, Natsuki Yamanobe, Kazuyuki Nagata, and Mathias Clerc. Customize function of single switch user interface for robot arm to help a daily life. In *Robotics and Biomimetics, 2008. ROBIO 2008. IEEE International Conference on*, pages 294–299. IEEE, 2009.

[225] Holly A Yanco and James Gips. Driver performance using single switch scanning with a powered wheelchair: robotic assisted control versus traditional control. In *TITLE Proceedings of the RESNA'98 Annual Conference: The State of the Arts and Science (Minneapolis, Minnesota, June 26-30*, page 309. ERIC, 1998.

[226] Chris Gibbons and Erin Beneteau. Functional performance using eye control and single switch scanning by people with als. *SIG 12 Perspectives on Augmentative and Alternative Communication*, 19(3):64–69, 2010.

[227] Susie Blackstien-Adler, Fraser Shein, Janet Quintal, Shae Birch, and Patrice L Tamar Weiss. Mouse manipulation through single-switch scanning. *Assistive Technology*, 16(1):28–42, 2004.

[228] Pradipta Biswas and Peter Robinson. A new screen scanning system based on clustering screen objects. *Journal of Assistive Technologies*, 2(3):24–31, 2008.

[229] Jutta Treviranus and Rosemary Tannock. A scanning computer access system for children with severe physical disabilities. *American Journal of Occupational Therapy*, 41(11):733–738, 1987.

[230] Sareh Rajaee. Web browser software with single switch interface for pdas or computers. In *Bioengineering Conference, 2004. Proceedings of the IEEE 30th Annual Northeast*, pages 253–254. IEEE, 2004.

[231] John R Wilson. *Virtual reality for industrial application: opportunities and limitations*. Hyperion Books, 1996.

[232] Miscrosoft Speech API. https://msdn.microsoft.com/en-us/library/jj127860.aspx. Accessed: 2017-4-1.

[233] Kate S Hone and Robert Graham. Towards a tool for the subjective assessment of speech system interfaces (sassi). 2000.

[234] Jakob Nielsen. Usability metrics: Tracking interface improvements. *Ieee Software*, 13(6):12, 1996.

[235] Michael Friendly and Daniel J Denis. Milestones in the history of thematic cartography, statistical graphics, and data visualization. *U RL http://www. datavis. ca/milestones*, 2001.

[236] Joe Hellerstein. Parallel programming in the age of big data. *Gigaom Blog*, 2008.

[237] Toby Segaran and Jeff Hammerbacher. *Beautiful data: the stories behind elegant data solutions*. " O'Reilly Media, Inc.", 2009.

[238] Martin Hilbert and Priscila López. The world's technological capacity to store, communicate, and compute information. *science*, 332(6025):60–65, 2011.

[239] What is big data? – bringing big data to the enterprise. www.ibm.com. Accessed: 2017-3-15.

[240] Mark A Beyer and Douglas Laney. The importance of 'big data': a definition. *Stamford, CT: Gartner*, pages 2014–2018, 2012.

[241] Danah Boyd and Kate Crawford. Six provocations for big data. In *A decade in internet time: Symposium on the dynamics of the internet and society*, volume 21. Oxford Internet Institute Oxford, 2011.

[242] Pak Chung Wong and R Daniel Bergeron. Multivariate visualization using metric scaling. In *Proceedings of the 8th conference on Visualization'97*, pages 111–ff. IEEE Computer Society Press, 1997.

[243] EAR Valiati. Taxonomia de tarefas para técnicas de visualização de informações multidimensionais. *Porto Alegre, PPGC/UFRGS, Recuperado el*, 5, 2005.

[244] Raquel M Pillat, Eliane RA Valiati, and Carla MDS Freitas. Experimental study on evaluation of multidimensional information visualization techniques. In *Proceedings of the 2005 Latin American conference on Human-computer interaction*, pages 20–30. ACM, 2005.

[245] Edward R Tufte. Envisioning information. *Optometry & Vision Science*, 68(4):322–324, 1991.

[246] Jacques Bertin. Semiology of graphics: diagrams, networks, maps. 1983.

[247] Benjamin Jotham Fry. *Organic information design*. PhD thesis, Massachusetts Institute of Technology, 2000.

[248] Dorina Marghescu. „multidimensional data visualization techniques for financial performance data: A review". *TUCS Technical Report No 810*, 2007.

[249] Alfred Inselberg. The plane with parallel coordinates. *The visual computer*, 1(2):69–91, 1985.

[250] Earliest known uses of some of the words of mathematics.

[251] Edward R Tufte. The visual display of quantitative information. *Journal for Healthcare Quality*, 7(3):15, 1985.

[252] Deborah F Swayne, Duncan Temple Lang, Andreas Buja, and Dianne Cook. Ggobi: evolving from xgobi into an extensible framework for interactive data visualization. *Computational Statistics & Data Analysis*, 43(4):423–444, 2003.

[253] Chris Stolte, Diane Tang, and Pat Hanrahan. Polaris: A system for query, analysis, and visualization of multidimensional relational databases. *IEEE Transactions on Visualization and Computer Graphics*, 8(1):52–65, 2002.

[254] Matthew O Ward. Xmdvtool: Integrating multiple methods for visualizing multivariate data. In *Proceedings of the Conference on Visualization'94*, pages 326–333. IEEE Computer Society Press, 1994.

[255] Çağatay Demiralp, Michael S Bernstein, and Jeffrey Heer. Learning perceptual kernels for visualization design. *IEEE transactions on visualization and computer graphics*, 20(12):1933–1942, 2014.

[256] Richard A Becker and William S Cleveland. Brushing scatterplots. *Technometrics*, 29(2):127–142, 1987.

[257] Mihael Ankerst, Stefan Berchtold, and Daniel A Keim. Similarity clustering of dimensions for an enhanced visualization of multidimensional data. In *Information Visualization, 1998. Proceedings. IEEE Symposium on*, pages 52–60. IEEE, 1998.

[258] Maximiliano Escudero, María Luján Ganuza, Dayanna Wilberger, and Sergio R Martig. Scatter plot 3d. In *IX Workshop de Investigadores en Ciencias de la Computación*, 2007.

[259] David S Ebert, Randall M Rohrer, Christopher D Shaw, Pradyut Panda, James M Kukla, and D Aaron Roberts. Procedural shape generation for multi-dimensional data visualization. *Computers & Graphics*, 24(3):375–384, 2000.

[260] Niklas Elmqvist, Pierre Dragicevic, and Jean-Daniel Fekete. Rolling the dice: Multidimensional visual exploration using scatterplot matrix navigation. *IEEE transactions on Visualization and Computer Graphics*, 14(6):1539–1148, 2008.

[261] D3.js library, released under the BSD license by Mike Bostock. https://d3js.org/. Accessed: 2017-4-1.

[262] QS world university ranking. http://www.iu.qs.com/university-rankings/world-university-rankings/. Accessed: 2017-4-1.

Appendix A

MoSCoW Questionnaire

MoSCoW Questionnaire

Specification	Comment	MSCW
1. Have your hands free while you run the maintenance procedure.	The maintainer has both hands dedicated to perform the maintenance task without interruption.	
2. Don't stop the maintenance procedure to obtain information.	It is not necessary to stop the maintenance procedure to use a tool such as a smartphone or a tablet where the maintenance application displays support information.	
3. View support information while performing the procedure, having your hands free.	The device allows you to view information while performing the procedure without interruption.	
4. The application provides real-time support.	Depending on the scene framed by the device, the application displays specific aid such as information about the operations to be performed and the objects involved by the procedure.	
5. The real-time support is textual.	The device displays the text of the operation to be performed.	
6. The real-time support is an audio.	A voice explains the procedure to perform depending on the framed scene.	

MoSCoW Questionnaire

Specification	Comment	MSCW
7. The real-time support is visual (indicators).	The objects/parts involved by the procedure are highlighted with arrows and/or graphical indicators of various kinds.	
8. The real-time support is visual (3D model).	The objects/parts involved by the procedure are highlighted by superimposing the real image with a 3D model of parts of the object.	
9. The real-time support is visual (3D animation).	The application superimposes on the real scene a 3D animation that simulates the operation to be carried out for the framed scene.	
10. The application can provide support also offline.	The instructions provided by the device do not depend on the scene framed and are viewable at any time (i.e. a previously recorded video).	
11. NOT real-time support is static.	Instructions are displayed as text and images showing the operations to be carried out.	
12. NOT real-time support is dynamic (virtual).	A video with a 3D representation of environment/equipment/items of interest. 3D animations show the operations to be performed.	
13. NOT real-time support is dynamic (augmented reality).	A video with a real representation of environment/equipment/items. 3D animations show the operations to be performed.	
14. NOT real-time support is dynamic (real).	A video where one maintainer carries out the maintenance procedure explaining all the steps.	

MoSCoW Questionnaire

Specification	Comment	MSCW
15. The application offers a menu with a list of available procedures	The maintainer selects the procedure he wants to get instructions or information about.	
16. The application allows you to interact with the video and the animations shown.	The maintainer can pause and restart a video/animation. This may lead to the suspension of the procedure.	
17. The application allows you to move through the various stages of a procedure.	The maintainer can review a particular step or skip operations that have already been carried out.	
18. The interaction with the device is physical (mouse/keyboard).	The maintainer interacts with the application through mouse and keyboard.	
19. The interaction with the device is physical (touch controls).	The maintainer interacts with the commands shown on the device screen, by touching the screen of the device.	
20. The interaction with the device is through control buttons on glasses	The maintainer interacts with the application by a set of control buttons on the glasses used to displayed augmented reality information	
21.The interaction with the device is through voice commands.	The maintainer interacts with the application saying the command to execute into a microphone.	
22. The device can be connected to a telephone network (i.e. 3G) to make phone calls.	The maintainer can make a call for instructions to an expert.	
23. The application can connect to a server remotely controlled by an operator who can provide assistance.	The maintainer can interact with the server to ask for additional information or clarification.	

MoSCoW Questionnaire

Specification	Comment	MSCW
24. The device can be connected to a telephone network (3G or 4G) to have Internet connectivity.	The maintainer may request assistance to a remote server that interacts with the application on the device if the place where the operation is carried out has telephone coverage.	
25. The device can be connected to a LAN, Wi-Fi or Internet network to connect to a server.	The maintainer may request assistance to a remote server that interacts with the application on the device if the place where the operation is carried out has LAN, Wi-Fi or Internet coverage.	
26. The application can interact with a Server by sending images.	The maintainer can send a picture to ask for clarification to an expert/technical manager about how to perform the maintenance procedure.	
27. The application can interact with a server receiving images, videos or 3D animations.	The maintainer may ask an expert/technical manager for the sending of clarification images, videos or 3D animations on how to perform the maintenance procedure.	
28. The application provides communication with the Server via text chat.	The maintainer may ask real-time questions to an expert/technical manager that operates the Server through written communication.	
29. The application provides communication with the Server via an audio link.	The maintainer may ask questions in real time talking to the expert/technical manager that operates the server.	
30. The application allows you to ask the Server instructions for a procedure not present on the terminal.	The maintainer may request the Server instructions for a new procedure not present on the terminal of the maintainer.	

MoSCoW Questionnaire

Specification	Comment	MSCW
31. The application and interface are in multiple languages.	The maintainer can choose the language he understands best.	
32. The instructions are displayed on binocular glasses.	The instructions are displayed on glasses for augmented reality, leaving your hands free.	
33. HaThe instructions are displayed on monocular glasses.	You are using a device with a single lens, leaving your hands free and allowing you to use other optical instruments.	
34. Use of glasses for augmented reality integrated into a helmet (Only for virtual training).	The maintainer wears a helmet equipped with LCD screens at eye level showing the surrounding environment and on which instructions are superimposed. The operator is unable to see directly the real world that surrounds him.	
35. Use of marker-less approaches.	The machine cannot be recognized by the AR tool by applying markers on its surface.	
36. The application recognizes the environment/the machine.	From the image/video the application is able to suggest a list of procedure to the operator.	
37. The application has video tracking capabilities.	i.e. the application tracks the finger to choose between items or buttons.	
38. The device supports different video formats (2D, side-by-side 3D).	Accordingly to the operator the video is shown in 2D or 3D.	
39. The device records a video.	During the operation the video is recorded for future activities.	
40. The application has layer management capabilities.	The application can show the whole machine, but also the inner layers.	

Appendix B

Augmented Reality SDK Comparison

AR SDK Comparison

SDK	Pricing	Marker Tracking	Natural Feature	Visual Search	Face Tracking	IMU Sensors	Content API	Framework	Unity 3D
3DAR	Free					✓			
ALVAR	Free + Commercial	✓	✓						
AndAR	Open Source	✓							
AR23D	Free + Commercial	✓							
ARLab	Free + Commercial	✓		✓	✓	✓			
ARmedia	Free + Commercial	✓							
ARMES	Commercial	✓	✓						
ARToolkit	Free + Commercial	✓	✓						
ArUco	Open Source	✓							
ATOMIC Authoring Tool	Open Source	✓					✓		
Aurasma	Free + Commercial		✓	✓					
BazAR	GNU License		✓						
Beyond Reality Face	Free + Commercial				✓				
Cortexica	Commercial			✓					
D'Fusion	Free + Commercial	✓	✓		✓				✓

AR SDK Comparison

SDK	Pricing	Marker Tracking	Natural Feature	Visual Search	Face Tracking	IMU Sensors	Content API	Framework	Unity 3D
Designers ARToolkit	Free	✓					✓		
DroidAR	Open Source	✓				✓			
FLARToolkit	Open Source	✓							
Goblin XNA	Open Source	✓					✓		
HOPPALA	Free						✓		
IN2AR	Free + Commercial		✓						
instantreality	Free + Commercial				✓				
iPhone ARKit	Open Source					✓			
IQ Engines	Free + Commercial			✓					
Kharma	Open Source						✓		
Kooaba	Free + Commercial			✓					
Koozyt	Commercial	✓							
layar	Free + Commercial		✓	✓		✓	✓		
LibreGeoSocial	Open Source					✓			
LinceoVR	Commercial		✓				✓		
Luxand FaceSDK	Commercial				✓				
Metaio SDKs	Free + Commercial	✓	✓	✓	✓	✓	✓	✓	✓
Minerva	Open Source	✓					✓		
mixare	Open Source								
MXR Toolkit	Open Source	✓							
NyARToolkit	Open Source	✓							✓
Obvious Engine	Commercial		✓						✓
OpenSpace3D	Open Source	✓					✓		
osgART	Open Source	✓	✓				✓		
PanicAR	Free + Commercial					✓			
PointCloud	Free + Commercial			✓			✓		
popcode	Commercial		✓						

AR SDK Comparison

SDK	Pricing	Marker Tracking	Natural Feature	Visual Search	Face Tracking	IMU Sensors	Content API	Framework	Unity 3D
PTAM	GNU License								
Qoncept AR	Commercial	✓	✓			✓			
Robocortex	Free + Commercial	✓	✓						
SLARToolkit	Open Source	✓							
SSTT	Open Source	✓	✓				✓		✓
String	Free + Commercial	✓							✓
Studierstube	Open Source						✓	✓	
UART	Open Source	✓						✓	
Viewdle	Commercial				✓				
Vuforia	Free + Commercial	✓	✓						✓
Wikitude	Free + Commercial	✓	✓			✓	✓		
windage	GNU License	✓	✓						
yvision	Free + Commercial	✓					✓	✓	
Zenitum Feature Tracker	Commercial	✓	✓						