POLITECNICO DI TORINO
Repository ISTITUZIONALE

An Augmented Reality System to Support Fault Visualization in Industrial Robotic Tasks

*Original*
An Augmented Reality System to Support Fault Visualization in Industrial Robotic Tasks / Avalle, Giancarlo; DE PACE, Francesco; Fornaro, Claudio; Manuri, Federico; Sanna, Andrea. - In: IEEE ACCESS. - ISSN 2169-3536. - ELETTRONICO. - 4:(2019), pp. 132343-132359. [10.1109/ACCESS.2019.2940887]

*Availability:*
This version is available at: 11583/2751052 since: 2019-09-19T17:48:37Z

*Publisher:*
IEEE

*Published*
DOI:10.1109/ACCESS.2019.2940887

*Terms of use:*

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

*Publisher copyright*

(Article begins on next page)

29 March 2024

IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received August 13, 2019, accepted August 29, 2019, date of publication September 11, 2019, date of current version September 25, 2019.

*Digital Object Identifier 10.1109/ACCESS.2019.2940887*

# An Augmented Reality System to Support Fault Visualization in Industrial Robotic Tasks

**GIANCARLO AVALLE[1], FRANCESCO DE PACE[1], CLAUDIO FORNARO[2], FEDERICO MANURI[1], AND ANDREA SANNA[1]**

[1]Dipartimento di Automatica e Informatica, Politecnico di Torino, 10129 Turin, Italy
[2]Facoltà di Ingegneria, Università Telematica Internazionale Uninettuno, 00186 Rome, Italy

Corresponding author: Francesco De Pace (francesco.depace@polito.it)

This work is part of the regional project HuManS (Human Centered Manufacturing Systems), and it was partially supported by Bando Piattaforma Tecnologica Fabbrica Intelligente POR FESR 2014/2020.

**ABSTRACT** The digitalization is transforming the very nature of factories, from automated systems to intelligent ones. In this process, industrial robots play a key role. Even if repeatability, precision and velocity of the industrial manipulators enable reaching considerable production levels, factories are required to face an increasingly competitive market, which requires being able to dynamically adapt to different situations and conditions. Hence, facilities are moving toward systems that rely on the collaboration between humans and machines. Human workers should understand the behavior of the robots, placing trust in them to properly collaborate. If a fault occurs on a manipulator, its movements are suddenly stopped for security reasons, thus workers may not be able to understand what happened to the robot. Therefore, the operators' stress and anxiety may increase, compromising the human-robot collaborative scenario. This work fits in this context and it proposes an adaptive Augmented Reality system to display industrial robot faults by means of the Microsoft HoloLens device. Starting from the methodology employed to identify which virtual metaphors best evoke robot faults, an adaptive modality is presented to dynamically display the metaphors in positions close to the fault location, always visible from the user and not occluded by the manipulator. A comparison with a non adaptive modality is proposed to assess the effectiveness of the adaptive solution. Results show that the adaptive modality allows users to recognize faults faster and with fewer movements than the non adaptive one, thus overcoming the limitation of the narrow field-of-view of the HoloLens device.

**INDEX TERMS** Augmented reality, collaborative robotics, fault visualization, industry 4.0.

## I. INTRODUCTION

With respect to the three past industrial revolutions, Industry 4.0 consists into a transformation process of the factory itself, from an automated facility to an intelligent one. An Industry 4.0 factory will be composed by several connected smart systems, capable of exchanging data and of taking actions and decisions autonomously. Since industrial robots are one of the industry key elements, they are strictly connected to this 4$^{th}$ industrial revolution and their positioning in this transformation process should be carefully planned. By analyzing the 2009 - 2014 period, it can be inferred a constant increase in the number of robots employed on the production lines [1], [2] reaching the 414.000 units in 2019 [3]. This fact, combined with the rapid transformation process of the Industry 4.0, may exclude human operators from the production process, generating undesired scenarios. Hence, it is necessary to develop new technologies that enhance the collaboration between humans and machines. The creation of such environments is a complex challenge. One of the main goals of the Human-Robot Collaboration (HRC) is to develop innovative interfaces that allow machines and human operators to collaborate in the same environment. There are several technologies that can foster collaboration between humans and machines [4]: among them, Augmented Reality (AR) seems to be a promising technology that can enhance the collaboration between industrial robots and human operators [5].

The first acknowledged AR prototype was developed by Sutherland [6] in 1968. Twenty years later, Milgram and Kishino formalized AR, by including it in the so called Mixed-Reality continuum [7]. Since then, technological

The associate editor coordinating the review of this manuscript and approving it for publication was Zhihan Lv.


This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see http://creativecommons.org/licenses/by/4.0/

improvements allowed the development of more sophisticated devices, capable of displaying 2D/3D high-definition virtual assets in the real environment. Three different visualization paradigms exist: handheld, projected and wearable see-through. Although handheld devices can exploit the relative high computational power of modern portable hardware (eg., smartphones and tablets) they lack usability, forcing users to keep their hands occupied. On the other hand, projected devices allow users to visualize virtual assets keeping their hands free. However, concerns arise when projecting on non-planar surfaces or in the occlusions management. Lastly, wearable devices combine the advantage of having the hands free of the projected devices to the portability of the handheld ones. Nevertheless, they still suffer for a very narrow Field-of-View (FoV). It is possible to find several models of wearable devices on the market. The most known are the Meta 2[1], the Microsoft HoloLens [2] or the Moverio BT-300[3].

In general, the AR market is increasing, passing from 5.91$ billion to more than 198$ billion[4] and since the AR technology should reach the plateau of productivity in 5-10 years[5], it is expected that it will be completely adopted in the Industry 4.0 domain.

AR shows its effectiveness in several industrial scenarios, such as maintenance and repair operations, inspection processes and to improve the collaboration with industrial machines [8]. Although AR is used in several ways with industrial robots (e.g., visualizing trajectories, intentions or the workspace of the manipulator [9]–[11]), very few works have tried to explore the idea of visualizing industrial robot faults by means of virtual metaphors. When a fault occurs on a manipulator, the robot is suddenly stopped to avoid any possible accident. Since in a human-robot collaborative scenario, human operators work side-by-side with industrial robots, an unexpected block of the robot's movements may increase stress and anxiety in the human operators because they are not aware of what is happening in the manipulator's controller. This work fits within this context and it proposes an AR system to display industrial robot faults. In particular, the whole design process is described from the choice of the graphic metaphors to the algorithm for placing these metaphors always in the best position with respect to the user.

This work is part of a regional project called HuManS: Human centered Manufacturing Systems, result of a collaboration among Politecnico di Torino and several IT companies, including COMAU, an Italian company leading of the industrial robots market. The main goal of this project is to develop innovative manufacturing paradigms, using a human-centered design approach. These new paradigms will make human operators capable of collaborating with industrial machines in smart factories, giving technicians the opportunity to take full advantage of new technologies and keeping them at the heart of the production process.

This paper is structured as follows: Sec. II presents the current state of the art concerning the use of AR technologies in the industrial robots' domain. It presents also a classification of the most common robot faults and the methodologies adopted to manage them. Section III gives an overview of the overall employed working flow. Section IV presents the procedure used to determine the virtual metaphors of the industrial faults. Section V illustrates the hardware and software architectures, along with the implementation of the proposed AR system. Section VI introduces the tests carried out to assess the effectiveness of the proposed solution. Section VII presents the related results. Finally, Sec. VIII details conclusions and possible future works.

## II. RELATED WORKS

### A. AUGMENTED REALITY AND INDUSTRIAL ROBOTS

AR can be employed in different ways to enhance the interaction and the collaboration with industrial robots. Several works [12]–[18] have explored the idea of visualizing the robot's path using AR systems. In [19], an interactive AR path generation system is proposed. Different interaction modalities have been developed: users can add, remove or modify spatial points directly in the real environment using a custom probe. Results show that the unskilled users have been able to program the robot in almost half of the time with respect to the normal teach-in methods. A similar work is proposed by Zaeh et al. in [20]. Users can interact with an AR projected path using a 6-Degree-of-Freedom (DOF) input pen. After the path creation, the motion of a virtual robot is shown on a desktop interface. Hence, it is possible to immediately verify if the generated path is collision-free and thus suitable for the real industrial manipulator. Outcomes reveal that the generation of a suitable path can be achieved in less than one fifth of the time required by a classic teach-in method. Concerning the use of AR wearable devices, an approach based on the Microsoft HoloLens device is proposed in [21]. Exploiting the Simultaneous Localization and Mapping (SLAM) of the HoloLens, the system is capable of recognizing an industrial robot arm and its surroundings, correctly aligning the corresponding virtual meshes. Then, users can add spatial points in the real environment using gaze, gestures and voice-based interactions. Despite results indicate that the proposed AR system seems to require a higher mental workload with respect to normal teach-in methods, the total time and the physical workload required to complete given tasks result to be lower.

Regarding the direct control of an industrial robotic arm, AR has been employed to get a feedback on an active user's input. Frank et al. [22] developed an handheld AR interface to control a 6-DOF robotic arm. Users can provide the robotic arm with the objects to be picked by selecting them on a tabletop User Interface (UI). Virtual meshes are

[1] http://www.metavision.com/
[2] https://www.microsoft.com/en-us/hololens
[3] https://tech.moverio.epson.com/en/bt-300/
[4] https://www.statista.com/statistics/897587/world-augmented-reality-market-value/
[5] https://www.gartner.com/smarterwithgartner/5-trends-emerge-in-gartner-hype-cycle-for-emerging-technologies-2018/

superimposed on the corresponding real objects to highlight the selection process. Users found very comfortable interacting with this system, showing to be able to precisely command the robot to move the real objects. Additional examples can be found in [23]–[26]. Su et. al [23] developed a tablet-based AR interface to control the end-effector (EE) of a virtual robotic arm. Two different interaction modalities have been implemented: in the first one, users can control the virtual EE by physically dragging the table, whereas in the second modality the EE can be controlled by using virtual arrows provided by the UI. In addition to control a robotic arm using a tablet device, Gradmann et al. [24] introduced an object recognition system based on a RGB camera combined with a depth one. When the system recognizes a set of real objects, some virtual meshes are superimposed on them to highlight the recognition procedure. Then, users can start a pick-and-place procedure using the information collected during the objects recognition process. Results show that 82% of the 111 experiments were performed correctly, showing the robustness of the proposed objects recognition process. In [25], a comparison between two different interaction paradigms using a wearable device to control a robotic arm is introduced. Users can control the EE of a robotic arm by using the head orientation or pointing gestures, both in combination with speech interface. Results suggest that the interaction by means of heading-based gestures requires less time and it is perceived as less mentally and physically demanding respect to the pointing-based gesture modality. In [26], an AR system to assist operators during collaborative 3D printing is presented. Using a custom AR wearable device, users can model different shapes in the real environment while a robotic 3D printer is printing the corresponding model.

It is worth to be noticed that AR technologies have been also employed to control robotic arms from remote locations ([27]–[36]). However, since this work fits within the context of a human and a robot that share the same environment, telerobotic works are out of the scope of this research.

Besides interacting with robotic arms, AR has been employed to visualize the workspace of industrial manipulators. In [37] a 3D sphere is centered at the robot's position and its radius varies according to the actual joints configuration. Alternatively to 3D AR, Vogel et al. [38] proposed a projected system to visualize static workspaces of high-payload robots. The system monitors the human operator's position by means of a tactile floor. Three different zones, called *free zone*, *warn zone* and *critical zone*, respectively, are projected on the floor using different colors. As the operator gets close to the critical zone, the manipulator's movements are slowed down until completely stopped when the operator enters the critical zone. The system has been improved in [39] to project dynamic zones that can change their shapes according to the robot's joints configuration. An analogous work is proposed in [11], in which a projected interface is used to delimit a low-payload industrial manipulator. Besides showing safety zones, the system is capable of detecting undesired violations by using a mask subtraction approach [40]. As soon as a violation is detected, the robot's movements are stopped and the color of the projected lines is changed to red to highlight the intrusion. Alternatively to the use of RGB cameras, Hietanen et al. [41] proposed a depth camera-based approach. Violations are detected by computing the difference between the original depth map and the current one. The system has been compared to a RGB approach to evaluate the total task time and the robot idle time. Results show a significant reduction for both task and robot idle times.

AR technology can be also employed to display task information or to highlight specific parts of an industrial manipulator. In [42], task instructions are displayed on a desktop interface to support human operators during human-robot collaborative assembly procedures. An AR projected interface to display interactive assembly instructions is proposed in [43]. A list of available programs is projected on a touch-enabled table. Users can add data to programs to fulfill task requirements. Malý et al. [44] developed a wearable AR system to enhance several components of a robotic arm using different 3D shapes. Qualitative results seem to indicate that the virtual shapes should be modeled using either an outlined or virtual no-transparency design. Independently of the visualization choice, the same design should be employed for the entire manipulator, thus avoiding mixing different styles. Several projects explored the idea of visualizing the robot's future intentions using AR interfaces. Virtual 3D arrows can be employed to display the future motion of the EE [10] or the movement of a virtual representation of an industrial manipulator can be visualized to improve the safety perception [45]. Further, projected AR interfaces can be used to visualize the objects that will be manipulated by the industrial robots. In [46] an object-aware dynamic projection system is proposed. The system is capable of tracking moving objects and projecting the future robot's working area. Furthermore, objects that will be manipulated by the manipulator are highlighted by superimposing on them 2D projected metaphors. Despite results demonstrate that the proposed AR interface has been preferred in terms of effectiveness and efficiency, it might not be enough to just show immediate intentions but it should also be considered the possibility of adding an overview of the overall task.

Given an overview on how AR technologies can be used with industrial manipulators, in the following the most common robot fault typologies and the current approaches to solve them are presented.

### B. ROBOT FAULTS

It is possible to classify faults in at least four different categories [47], [48]:

- faults on the sensors: even if the physical quantity measured by the sensor is not affected by errors, the measured values can be incorrect;
- faults on the actuation system: motors and motor drivers can be damaged;

**TABLE 1. An example of fault log file. The *alarm_count* column shows how many times an error has occurred. The second and third columns indicate the id of the robot and the fault, respectively. The fault severity is expressed with an integer number between 1 and 11 (*severity* column). The *alarm_text* column provides a brief description of the related fault.**

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

- faults on the mechanical structure: mechanical components, such as joints, can stop moving due to brake faults or collisions;
- overloading problems: due to the intrinsic raising limits of manipulators, problems may occur if the robot tries to raise up too heavy objects.

When one or more faults occur, the correct functioning of the manipulator is compromised and thus its movements and activities are stopped for safety reasons. Since this behavior may greatly affect factory's production [49], successful fault solving strategies should be pursued. Nowadays, one common procedure used to manage and solve industrial robot faults is represented by the following workflow:

- the manipulator's activities are stopped;
- the fault's description is saved in a log file (see Table 1);
- technicians, checking the log file, try to cope with the error using their experience and technical manuals.

This procedure may result unsatisfactory for at least two different reasons. Firstly, time required to solve the problem strongly depends on the clarity of the log file description and on the technician's experience. Secondly, since human operators cannot be immediately informed on the robot's internal status, their trust in the industrial manipulators may decrease, compromising the human-robot collaborative context. Hence, it results necessary to find alternative solutions to manage and solve robot faults. AR can be suitable to achieve this purpose. Exploiting its capability of enriching the real environment with virtual information, faults can be clearly visualized either on virtual Graphical User Interfaces (GUIs) or on the manipulator itself, allowing users to be immediately informed about the internal robot's state. To the best of the authors' knowledge, only few works have explored the idea of visualizing robot faults using AR interfaces. In [50], a desktop interface is employed to visualize the measurements of some robot's sensors. Two different virtual graphs are shown to get an immediate feedback on the sensors' state. Although authors clearly explain the benefit of such typology of visualization, the system has not been tested and thus it is not possible to verify its effectiveness. Another work has been proposed in [51]. It consisted in an attempt of visualizing faults both on a virtual manipulator and on a real humanoid robot. Despite collected results suggested that some virtual metaphors represent faults better than others, the limited FoV of the adopted wearable device has greatly compromised the effectiveness of the proposed AR interface. The lack of a mechanism that adapts the metaphors' positions, orientations and sizes according to the employed device did not allow users to clearly visualize the proposed virtual metaphors.

The virtual assets positioning may greatly affect the UI's effectiveness. Works in [52], [53] present two different approaches to find the best position for a projected AR interface for a human-robot collaborative task. In [52], the position and orientation of a human worker are continuously tracked and thus the system is able to project an AR interface in a position that is always visible from the user. Conversely, Claassen et al. [53] proposed a system capable of finding the most suitable planar surface to project an interactive AR interface to control a mobile industrial manipulator. User's input are detected using a background subtraction strategy introduced in [54]. Results indicate that the projection angle should be kept around 30° to obtain a satisfactory user's input detection rate. The main limitations of these two works can be summarized as follows:

- the lack of tests in [52] makes impossible to ascertain the system usability;
- in [53], the proposed adaptive interface was not compared with a non adpative one. Moreover, it is not clear its behavior when no suitable planes are detected.

On the other hand, this paper presents an adaptive AR system that is capable of displaying industrial robot faults in real-time, independently of the surroundings. Hence, faults are always clearly recognizable by the users, reducing the overall fault management's times and costs. The placements of the faults are computed according to the following parameters:

- the reference object;
- the user's position and orientation;
- the FoV of the employed device.

In order to properly manage robot's errors, technicians need to clearly visualize the manipulator's parts affected by faults. Augmented assets should be positioned in areas close to the fault's location without occluding the manipulator itself. Thus the proposed system adapts the assets placements according to the user's movements, keeping the virtual metaphors always visible, correctly oriented and properly scaled. An image segmentation algorithm is presented to position the 3D metaphors in areas not occluded by the industrial manipulator. In this way, technicians can immediately identify faults, thus reducing repairing times and costs. In order to evaluate the effectiveness of the proposed solution, a comparison with a non adaptive metaphors placement strategy (such as the one proposed in [51]) has been performed.

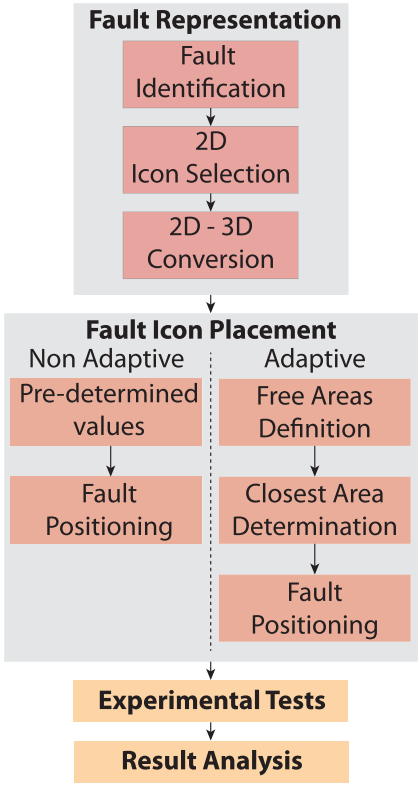



**FIGURE 1.** The working flowchart.

## III. THE WORKING FLOWCHART

The overall working flowchart is composed by four different steps (see Fig. 1):

1) Fault Representation (FR);
2) Fault Icon Placement (FIP);
3) Experimental Tests (ET);
4) Result Analysis (RA).

The FR step concerns the strategy adopted to figure out which 3D metaphors best represent industrial robot faults. Starting from the identification of the most common faults, a set of 2D icons has been determined in order to represent them. Then, the 2D set has been converted to 3D representations to be employed in the AR system. In the FIP step, two different modalities of placement (non adaptive and adaptive) of the 3D virtual metaphors have been developed. The first one relies on the use of pre-determined parameters, whereas the second one dynamically places the assets in the real environment. Specifically, starting from the definition of the areas not occupied by the manipulator in the FoV of the AR device, the adaptive modality determines the closest free area to the fault location, visible by the user. In order to assess the effectiveness of the adaptive modality, a comparison with the non adaptive one is carried out in ET step. Finally, the gathered results have been analyzed in the RA step.

In the following sections, each step is presented in the detail.

**TABLE 2.** The 10 identified base sentences.

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

## IV. FAULT REPRESENTATION

In order to improve the fault management procedure illustrated in Sec. II-B, a graphical representation of the faults should be added to make immediately clear what kind of error occurred in the manipulator. Since faults are commonly represented using 2D icons (for instance, motor vehicle faults, smartphone battery warning, etc.), it has been decided to determine the 3D virtual metaphors by identifying a set of 2D icons that evokes as much as possible faults on industrial manipulators. Then, due to the specific AR device employed, a 3D representation has been preferred with respect to the adoption of a traditional 2D UI composed by the 2D icon dataset. In fact, the employment of 2D images positioned on planes different from those of the real objects may cause focal problems [55]. Furthermore, "*the more (2D) UI elements in screens space exist, the less attention is given to the AR space*" [56]. Hence, the selected 2D icons have been converted in 3D models. In the remaining of the section, the entire procedure is presented.

In order to figure out which 2D icons best represent industrial manipulator faults, a rigorous design approach has been adopted:

1) semantic analysis: extraction of the most relevant terms regarding industrial robot faults and generation of a 2D icon dataset;
2) 2D icon design: re-design process of the 2D icon dataset;
3) 2D icon selection: selection of the most representative 2D icons.

In the following, each step is introduced and presented.

### 1) SEMANTIC ANALYSIS

In order to realize which are the most frequent robot faults and how they are described in research literature, a semantic analysis approach has been adopted. Starting from [47], [48], the most significant sentences related to robot faults have been manually extracted, identifying ten different fault typologies called *base_list_sentences* (Table 2).

Furthermore, it has been verified that these ten typologies were mentioned in a fault log file provided by COMAU (this log file covers a period of two years). Then, the following procedure has been applied to each element of the *base_list_sentences* set:

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

**TABLE 3.** The *synonym_list_sentences* set. The word *fault* has two synonyms (in yellow). The word *speed* has three synonyms (in green). No synonyms have been found for *reducer*.

| | | |
|---|---|---|
| fault | speed | reducer |
| fault | velocity | reducer |
| fault | speeding | reducer |
| fault | hurrying | reducer |
| defect | speed | reducer |
| defect | velocity | reducer |
| defect | speeding | reducer |
| defect | hurrying | reducer |
| flaw | speed | reducer |
| flaw | velocity | reducer |
| flaw | speeding | reducer |
| flaw | hurrying | reducer |

In order to explain the above procedure, the fifth sentence of Table 2 will be used as an example. Firstly, from the S1 step, the sentence *fault speed reducer* has been obtained as output. Then, in the S2 step, a set of new sentences (henceforth called *synonyms_list_sentences*), including the starting one, has been generated using the synonyms of the sentence words. Synonyms have been obtained using the WordNet online tool[6] (Table 3 shows the *synonyms_list_sentences*). Finally, the generation of word permutations has been applied in S3 step. Each element of the *synonyms_list_sentences* has been considered as a set composed by $n$ elements that has been employed to generate all the possible unordered subsets formed by $k$ items, with $1 \leq k \leq n$. The number $T$ of unordered subsets at a specific $k$ is computed as:

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

where $n$ and $k$ are the number of words of a specific sentence and the dimension of a specific subset, respectively. Each subset has been used to query the online icon database TheNounProject[7]. This database provides over than two million of 2D icons and APIs to access images from scripts. Every time an icon was found, it was saved in the corresponding fault category.

Since icons have been collected using an automatic procedure, the final dataset was characterized by a huge variety of different styles. Hence, icons have been firstly subjected to a re-design process to homogenize the dataset and then converted in 3D virtual assets. These procedures are introduced in the following sections.

### A. 2D ICON DESIGN

Due to the huge number of collected icons, images not strictly related to the ten fault categories have been discarded. The final dataset was composed by 121 icons, not uniformly subdivided in the ten faults categories (see Table 4).

Despite icons came from the same database, they could have been designed by several distinct authors using different styles. Hence, a re-design process has been employed to homogenize the collected dataset. Analyzing the current state of the art related to icon design [57]–[59], one of the most effective representation of a 2D icon can be achieved using:

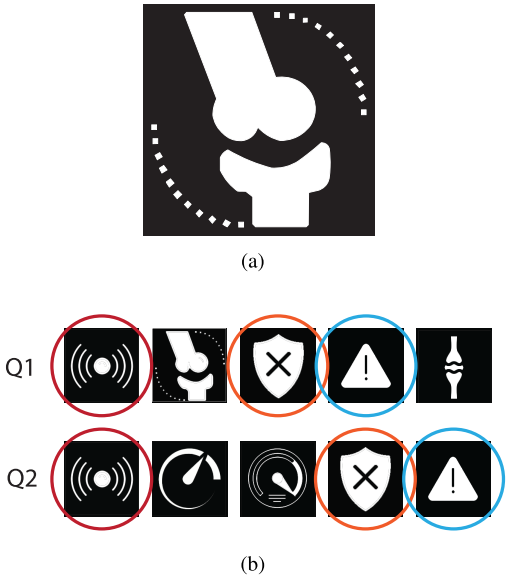
- plane composition;
- negative polarity;
- border.

Thus, all the collected icons have been converted following the above guidelines (an example of re-designed icons can be found in Fig. 2a).

[6]http://wordnetweb.princeton.edu/perl/webwn
[7]https://thenounproject.com/

**TABLE 4.** The number of collected icons, divided by category.

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




(a) (b)

**FIGURE 2.** A: an example of re-designed 2D icon. B: an example of icons that appear in different categories.

### B. 2D ICON SELECTION

A questionnaire has been submitted to both COMAU technicians (12 users) and Politecnico of Torino students (52 users) to figure out which icons best evoked a specific typology of fault. The questionnaire was composed by ten different questions, corresponding to the ten fault typologies. In each question, users had to indicate which icons were more suitable to represent manipulator faults by selecting one icon among those proposed. It has been also added the option ''none of these'' to indicate that none of these icons was suitable to

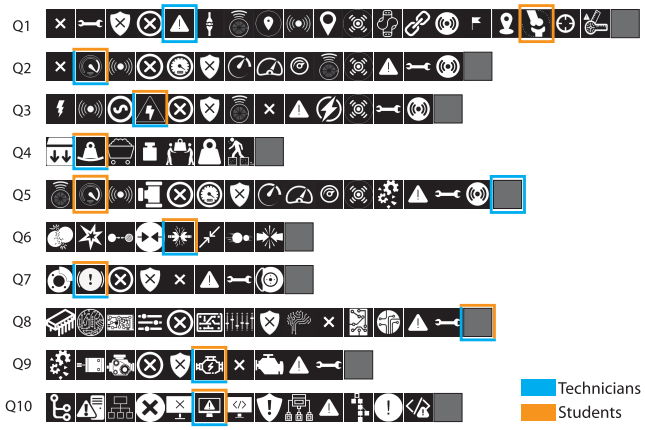



**FIGURE 3.** The questionnaires results. The mixed light blue-orange squares are used to indicate that technicians and students have selected the same icon.

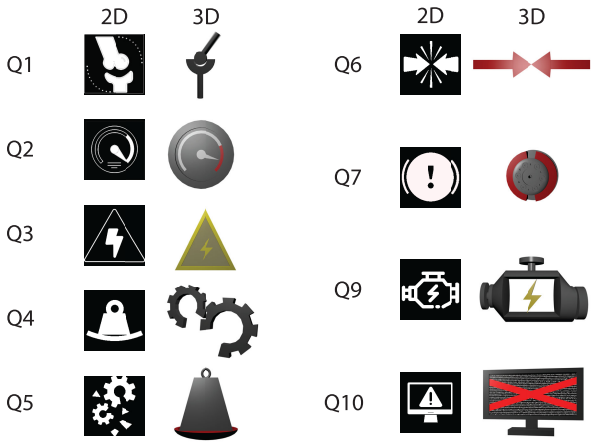



**FIGURE 4.** The 2D selected icons and their 3D representations.

represent the specific fault. Moreover, a form for comments and feedback has been provided for each question. Since the elements of the *base_list_sentences* have some common terms (e.g., the terms "fault" or "sensor" occur multiple times in different sentences), the same icon could appear several times in different categories (see Fig. 2b), representing two or more faults and thus compromising the effectiveness of the icon itself. As stated in [60], if the same icon expresses various and different meanings, the icon results to be ambiguous and users may not be able to understand the real meaning of the icon. Hence, to avoid any form of ambiguity, a set of *intra*-category rules has been employed to decide which 2D icons represent the fault typologies. In order to clarify the rules, let $S$ be an icon chosen from students, $T$ an icon selected by the technicians, $i$ an integer number representing a specific fault category, $c$ a comment, *None* the "none of these" option and *Null* to indicate that comments are not provided (see Pseudocode 1).

**Pseudocode 1** The *Intra-Category* Rules. Notice That if Both Technicians and Students do Not Select an Icon (Neither Express a Comment), the Category is Discarded

```
if T_i != None then
    return T_i
else if T_i(c) != Null then
    return T_i(c)
else if S_i != None then
    return S_i
else if S_i(c) != Null then
    return S_i(c)
else
    return -1
end if
```

As can be inferred from the intra-category rules, priority has been given to the technicians' choices. Once icons have been identified in each category, they have been analyzed to find out any possible repetitions. Since the number of possible cases can make arduous developing an automatic procedure but at the same time the number of categories is quite low, it has been decided to manually manage any possible conflict in the selection of the icons.

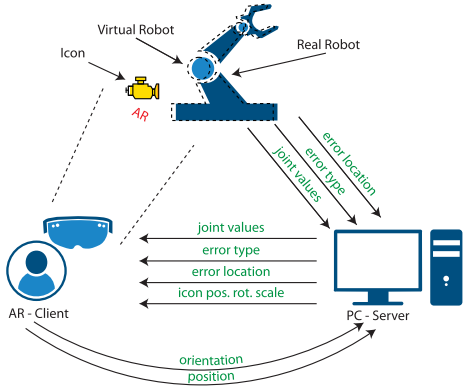



**FIGURE 5.** The hardware architecture.

The final set of 2D icons is presented in Fig. 4 2D-column. Because neither COMAU technicians nor Politecnico students have indicated a clear preference for Q8, this typology of fault has been discarded. Finally, a graphic designer has converted the 2D icons to 3D virtual animated assets (see Fig. 4 3D-column). Animations have been added to better express their semantic meaning. This new set of 3D icons has been employed in the interface of the AR client. In the following sections, the AR interface and the algorithm running on the server are presented.

## V. FAULT ICON PLACEMENT

In the following sections, the overall proposed system to position the 3D virtual metaphors is presented. Specifically, the hardware/software architectures and the AR interface are detailed. Finally, the adaptive/non adaptive solutions are introduced and discussed.

On the other hand, the software architecture is represented in Fig. 6, which shows the software details of the industrial manipulator, the AR client and the server.

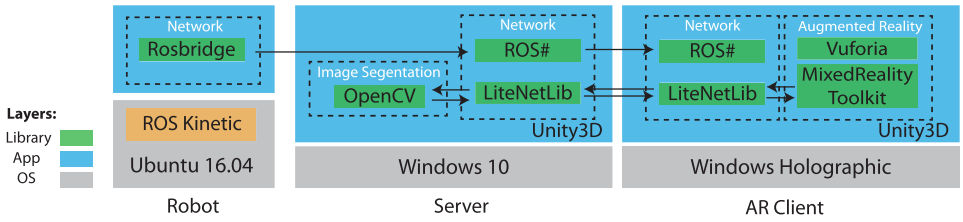



**FIGURE 6.** The software architecture.

### A. THE ARCHITECTURE

The hardware architecture is composed by three main actors: the industrial manipulator, the AR client and a remote server. The industrial manipulator is represented by the Niryo One arm robot[8]. Since it consists of a 6 degree-of-freedoms (DOFs) robotic arm, it can be considered suitable for representing configurations of an industrial manipulator. The AR client is represented by the Microsoft HoloLens, a Head-Mounted Display (HMD) device. A wearable device has been preferred respect to a handheld or projected solution. Therefore, technicians can keep their hands free to perform any possible tasks. Moreover, the visualization of the icon is independent from the environment (with a projected solution, assets may not be properly displayed if the projection surface is not planar). The server is represented by a desktop Personal Computer (PC) that runs an algorithm capable of acting in two different modalities: adaptive and non adaptive. Depending on the modality, the server dynamically places the 3D metaphors or it uses some pre-defined values.

All devices are connected on the same Local Area Network (LAN) using UDP socket connections. Since the image segmentation algorithm has been developed using image processing libraries that can require a high computational power, a client-server architecture has been employed to limit the battery consumption and to lower the HoloLens computational effort. Figure 5 shows the hardware architecture.

### B. IMPLEMENTATION

The algorithm running at the server side is capable of acting in two different modalities: adaptive and non adaptive. Independently of the employed modality, the server receives from the real robot:

- error location: an integer number corresponding to the joint affected by the fault (values range from 0 to 5);
- error type: an integer number corresponding to the fault typology (values range from 0 to 8);
- joint configuration: the values of the robot joints.

The server reads the joint configuration by subscribing to the joint_state topic, whereas the location and type errors are acquired by directly analyzing the stream socket data. Exploiting these data, the server is capable of generating a faithful representation of the actual state of the manipulator. Joint values are used to animate a virtual representation of the manipulator whereas the error data are used to render the corresponding 3D fault icon. In addition, the server sends to the AR client:

- joint configuration;
- error location and type;
- position, orientation and scale factor of the 3D icon to be aligned with respect to the real robot.

Joint configuration is forwarded to the AR client to animate a virtual mesh of the manipulator. Although this mesh is not visible to users, joint configuration is used to highlight the joint affected by the fault (see Sec. V-C).

If the server is acting in adaptive modality, it receives from the AR client the following data:

- the user's position: 3 floating values;
- the user's orientation: 3 floating values.

These values are sent every 20 ms and they are used to animate a virtual camera that represents the user's actual position and orientation. The server virtual camera's settings (FoV, far and clipping planes and resolution) have been set equal to the ones of the AR client virtual camera. Given these data, the server is capable of visualizing the virtual manipulator from the user's point of view (in terms of position, orientation and FoV). All data but joint values are serialized and deserialized using the Google Protocol Buffer[9].

### C. THE AR INTERFACE

The AR interface has been designed to assist technicians that work close to industrial manipulators. When a fault occurs on the robot, the robot's movements are suddenly stopped and technicians may not be aware of what is happening in the manipulator. Moreover, at the fault time, users may not have their attention paid to the manipulator. Hence, a combination of virtual assets and sound is employed to draw their attention toward the robot. The virtual assets are the following:

- a virtual representation of the Niryo robot;
- a virtual arrow;
- the virtual fault icons.

When the application starts, the virtual robot is superimposed on the real one using the tracking data acquired by detecting the image target, positioned on a table at a predefined distance from the real manipulator. The 3D mesh is kept invisible to not occlude the real robot. Thanks to the tracking information provided by the Vuforia SDK (see Fig. 6), both position and

[8]https://niryo.com/

[9]https://developers.google.com/protocol-buffers/

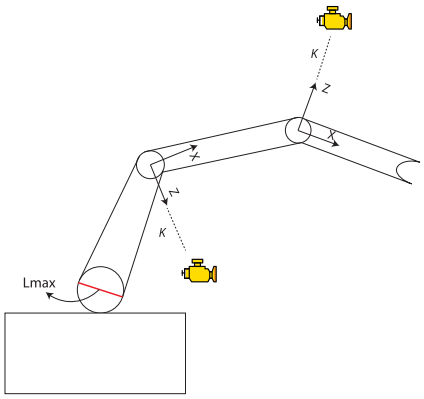



**FIGURE 7.** In the NA modality, the virtual icon is positioned at a predefined distance *k* along the joint local *z* axis.

orientation of the HoloLens are known. Thus, it is possible to figure out the technician's position/orientation with respect to the robot. These values are sent to the server on the socket connection every 20ms. If the technician is not gazing at the manipulator at the fault time, a virtual arrow (Fig. 8a) focuses the user on the joint affected by the fault. When the user is close enough to the manipulator to clearly recognize the joint, the 3D arrow disappears. In order to highlight the joint affected by the fault, the related 3D mesh is displayed (keeping invisible all the other joints of the manipulator) and its color is changed to red to emphasize the fault (see Fig. 8b). Furthermore, a sound alarm is played to capture the operator's attention. Depending on whether the algorithm running on the server is acting in adaptive or non adaptive modality, the fault virtual icon is rendered taking into account the user's movements and the device's FoV or using predefined scaling, position and orientation values.

### D. THE SERVER

The server is in charge of deciding the position, orientation and scale of the 3D fault icons. It can act in adaptive (A) or non adaptive (NA) modality. In the following sections, the two distinct modalities are presented.

#### 1) NON ADAPTIVE MODALITY

In the NA modality, some predefined values are sent to the AR client. Concerning the position value, the icon has been placed at a predefined distance $k$ along the Z direction of the local reference system of the joint affected by the fault. The $k$ constant has been defined as:

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

where $L_{max}$ represents the diameter of the manipulator's larger joint (see Fig. 7), equal to 12 cm. Therefore, the virtual icon can be placed close to the robot, but avoiding to be positioned inside the related joint. To calculate the dimensions of the virtual icons, it has been firstly determined the minimum distance $D_{robot}$ necessary to entirely visualize the manipulator using the HoloLens device, equal to 1.4 m. Then, some tests have been carried out to visualize the icons from that distance and it has been decided to employ icons with variable dimensions on the three axis ($x$, $y$, $z$) between 1 and $k$ cm. It is worth to be noticed that the each axis dimension is related to the shape of the icon (e.g., the joint position icon's height is larger than its width, etc.). Finally, once the size of all the icons had been established, a constant and equal scaling factor $S_c$ was assigned for all of them. Regarding the orientation, it is kept constant along the three axis, equal to (0, 0, 0).

#### 2) ADAPTIVE MODALITY

In the A modality, the server tries to place the 3D icon in a position close to the joint, always visible to the user and not occluded by the real manipulator. Exploiting the position and orientation information of the HoloLens, the server is able to visualize the robot from the same point of view of the user (Fig. 8c - 8d). This information is used in the following algorithm that can be divided into three distinct steps:

1) A1: determination of the icon's scale factor;
2) A2: determination of the icon's 3D position;
3) A3: determination of the icon's orientation;

Regarding A1, when the robot's controller notifies the server of a new fault, the server instantiates the corresponding virtual icon at the position of the related joint, using a pre-defined scale factor, equal to $S_c$ (see Sec. V-D.1). Since the icon's correct position has not been determined yet, the icon mesh is kept invisible. The scale value is updated at each frame using the following approach. Let $J(x_j, y_j, z_j)$ be the position of the joint affected by the fault and $V(x_v, y_v, z_v)$ the position of the virtual camera, both defined in the world reference system. The distance $D_{JV}$ is used to compute the scale factor $S_{icon}$ of the icon:

$$S_{icon} = (\frac{S_c}{D_{robot}}) * D_{JV} \tag{3}$$

In order to not excessively scale the icon, $S_{icon}$ is kept constant if $D_{JV}$ is minor or greater than two pre-defined values $D_{min}$ and $D_{max}$. $D_{min}$ and $D_{max}$ should be determined considering the working area, for the proposed use case they are equal to 35 cm and 3.5 m, respectively. Concerning the icon's position (A2 step), the subsequent procedure is applied when the camera movement exceeds some predefined values (in terms of position and orientation displacements):

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

Starting from B1, the 3D icon's bounding-box ($BB_{3D}$) is projected onto the camera screen space using the world-screen transformation matrix to compute a $BB_{2D}$ in pixel dimensions. $BB_{2D}$ approximates the space occupied by the icon in the camera screen space. Given $BB_{2D}$, the image acquired by the camera is analyzed and processed in order

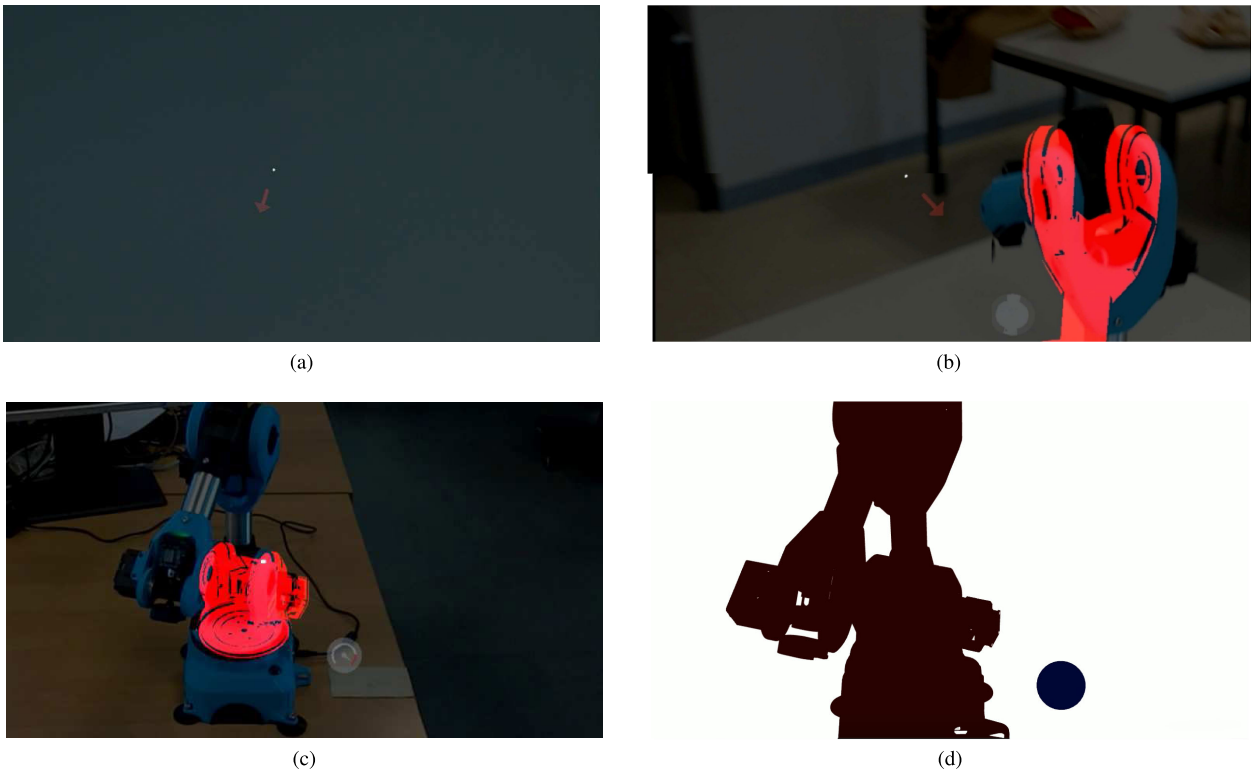



**FIGURE 8.** A: the virtual arrow used to capture the technician's attention. B: the joint affected by the fault is highlighted in red color. C - D: the same scene viewed from the user and server points of view, respectively.

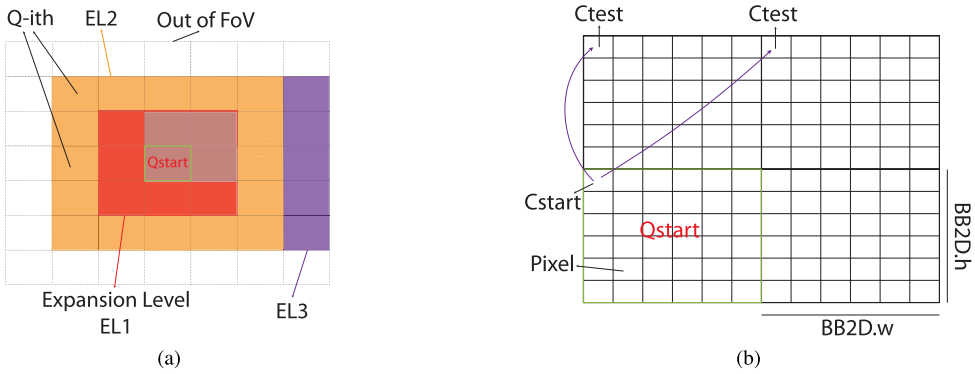



**FIGURE 9.** A: the *Q*s grid. EL1, EL2 and EL3 represent three different expansion levels (EL3 is the maximum expansion level). Since the dotted *Q*s are outside of the camera FoV, they are not considered in the proposed procedure. B: a zoomed view of the four grayed *Q*s of A. Each *Q* represents a pixel matrix, defined by $C_{start}$.

to find a suitable area, large enough to contain the icon and not occluded by the manipulator (B2 step). The proposed approach is based on the thresholding of the camera image using color information. In order to detect the areas occupied by the manipulator on the camera image, the color of the robot's virtual model has been set to red and the camera background to white. Hence, all the pixels of the camera image coloured in red represent forbidden spaces whereas white pixels identify possible suitable areas. Let $I_{rgb}$ be an $M$x$N$ matrix representing the RGB image acquired by the virtual camera, with $M$ and $N$ equal to the height and width resolution of the virtual camera. $I_{rgb}$ is then converted in the hue, saturation, value (HSV) color space, obtaining $I_{hsv}$. Applying an image segmentation algorithm on $I_{hsv}$, the thresholded matrix $I_t$ is obtained as:

$$I_t(x,y) = \begin{cases} 255 & r_{min} <= I_{hsv}(x,y) <= r_{max} \\ 0 & otherwise, \end{cases} \quad (4)$$

where $r_{min}$ and $r_{max}$ represent two constant values used to identify the red color and $I(x,y)$ identifies the value of a specific pixel in the corresponding matrix. Then, $I_t$ has been subdivided in quadrants $Q$s of dimensions equal to $BB_{2D}$, generating a grid (see Fig. 9a).

Given a starting quadrant $Q_{start}$, the main goal of the B3 step is to find a suitable $Q$ that minimizes the distance from $Q_{start}$.

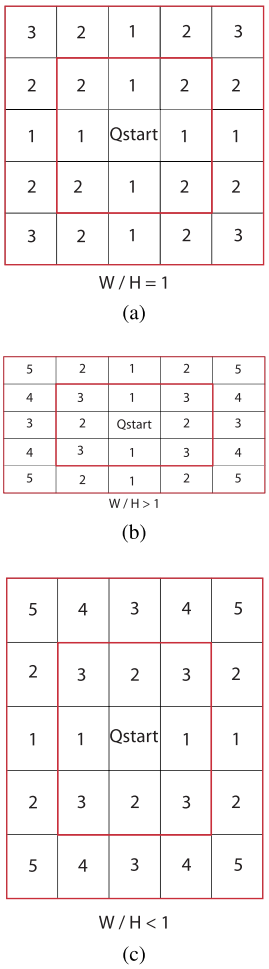



**FIGURE 10.** **The three distinct checking orders depend on the ratio between Qs width and height.**

Let $EL_{max}$ be the maximum *Expansion Level* (Fig.9a) determined as:

$$EL_{max} = max(Q_{r_max}, Q_{c_max}), \tag{5}$$

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

$$C_{test} = (C_{start}.x + k * BB_{2D}.w, C_{start}.y + u * BB_{2D}.h), \tag{6}$$

where $k$ and $u$ represent two integer numbers used to access the $Q$s of the grid and $C_{start}$ is the $(x, y)$ coordinate of $Q_{start}$ upper-left pixel (see Fig. 9b). In order to iterate over all the available $Q$s, the following equation has been employed to determine the $k$ and $u$ values of Eq. 6:

$$k = \pm e, u = -e + i, e - i, \tag{7}$$

where $e = 1, ..., EL_{max}$ and $i = e, ..., 0$ are positive natural numbers. Depending on the ratio between $BB_{2D}.w$ and $BB_{2D}.h$, three different checking orderings exist (Fig. 10).

Given an ordering, the algorithm iterates over all the $Q$s of a specific $s$ by first assessing those that are at a shorter distance from $Q_{start}$. The assessment order of different $Q$s that are at the same distance from $Q_{start}$ has been defined in advance. During the iteration, the algorithm checks whether a quadrant is suitable for positioning the 3D icon or not. The evaluation process consists of evaluating the sum of all the $I_t(x, y)$ of a specific $Q$. If the sum is greater than zero, some red pixels have been found and thus a part of the manipulator is occluding that specific $Q$. Otherwise, a suitable quadrant $Q_{selected}$ has been found and it will be used to position the 3D icon in the virtual environment.

Finally, starting from $Q_{selected}$, it is necessary to retrieve a position close to the fault location in the 3D space (B4 step). Let $V$ and $J$ be two positions representing the camera and the joint affected by the fault, both in world coordinate reference system. Using these positions, the following approach has been adopted (Fig. 12):

1) a ray is casted from the center of $Q_{selected}$;
2) determining a ray point $R$, the vector $\vec{VR}$ is computed;
3) the vector $\vec{VJ}$ is projected on $\vec{VR}$, finding a 3D position not occluded and close to the manipulator.

Once a suitable 3D position is determined, the icon's orientation is changed so the forward vector of its locale reference frame points to the camera position, keeping the icon always visible from the user point of view (A3 step). A more detailed pseudocode of the algorithm is shown in Appendix -A.

The proposed algorithm can be employed starting from a generic $Q_{start}$. For the proposed use case, the choice of $Q_{start}$ is dynamically computed, considering the changing in position and orientation of the user. To determine $Q_{start}$, the subsequent assumptions have been made:

1) if the icon has not been positioned yet (e.g., at the system's start-up) or it is no more in the FoV (e.g., rapid changes of the user's orientation), it is selected the $Q$ that encompasses the projection of the 3D position of the joint affected by the fault on the camera space (Fig. 11a - 11b);
2) if the icon is in the FoV, the $Q$ that encompasses the projection of the 3D position of the icon on the camera space is selected.

In the second case, an additional check is performed to verify whereas $Q_{start}$ is occluded or not by the manipulator. If the occlusion is verified, the algorithm does not look for the first adequate $Q$ of a specific Expansion Level $EL$ but it determines $N$ adequate quadrants ($Q_{potential}$) of $s$ ($N >= 1$) and it is chosen as $Q_{selected}$ the one that minimizes the Euclidean distance from the quadrant that encompasses the projection of the 3D position of the joint affected by the fault ($Q_{joint}$). Hence, abrupt and undesired movements of the 3D icon are avoided and the icon is kept as close as possible to the fault location (see Fig. 11c - 11d). Finally, the overall algorithm is not applied if the joint affected by the fault is not in the FoV of the employed device.

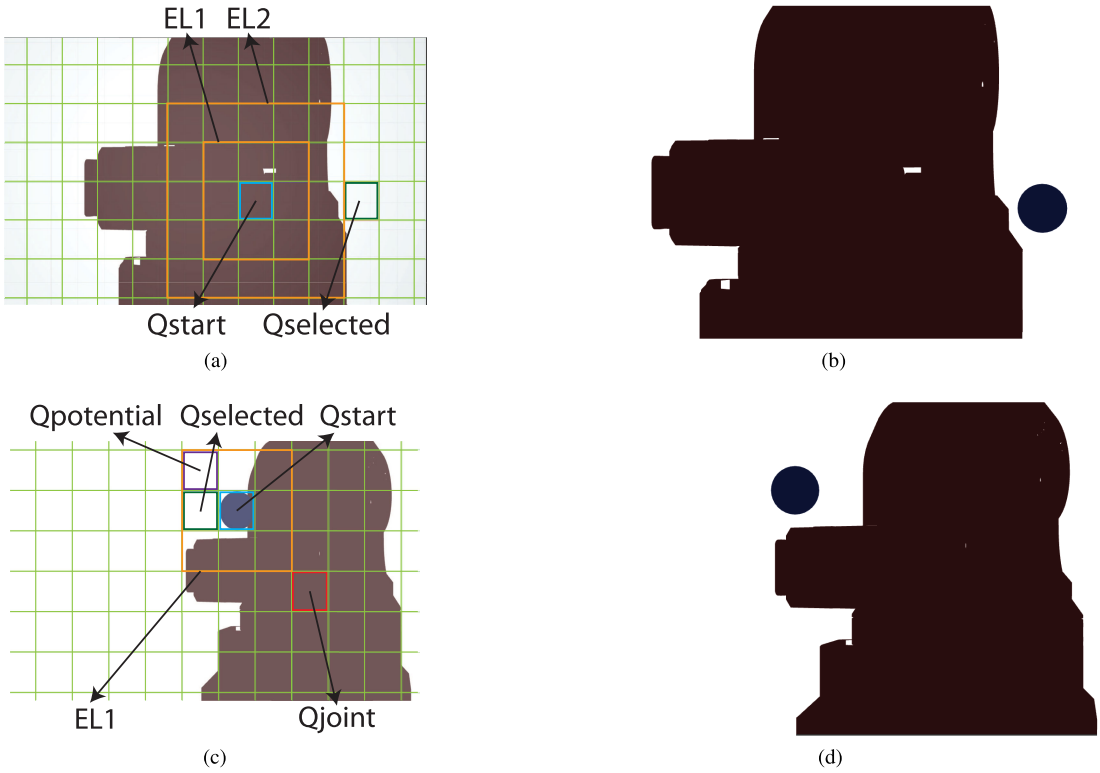



**FIGURE 11.** A: as $Q_{start}$ (in light blue) it has been selected the $Q$ that encompasses the projection of the position of the joint affected by the fault. B: the positioning of the 3D icon using the $Q_{selected}$ in A. C: as $Q_{start}$ it has been selected the $Q$ that encompasses the projection of the position of the icon. Since $Q_{potential}$ (in purple) is further from $Q_{joint}$ than $Q_{selected}$, it has been discarded. D: the positioning of the 3D icon using the $Q_{selected}$ in C.

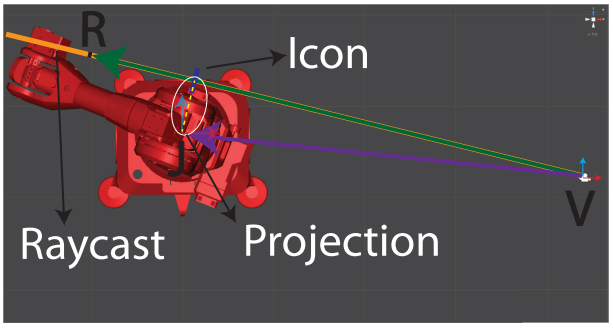



**FIGURE 12.** The projection of the $Q_{selected}$ x and y coordinates in the 3D space. The ray-cast is represented by the orange line. $V$ is the camera and $J$ represents the position of the joint affected by the fault.

## VI. EXPERIMENTAL TESTS

In order to compare the A and NA modalities, some tests have been carried out at Politecnico di Torino. Thirty-four people, with ages that ranged from 19 to 30 years, have evaluated both modalities, alternating every time the A-NA, NA-A orders to limit learnability effects. Participants were all volunteers and they gave written informed consent in accordance with the Declaration of Helsinki[10].

Tests have been structured to emulate a fault condition during a human-robot collaborative scenario, taking into account also situations in which the human operator is not giving attention to the manipulator (i.e., the manipulator is not in the FoV of the user):

- a set of 9 starting positions (SPs) has been defined. The SPs have been determined considering both locations near and far from the manipulator;
- user starts the experiment in a SP, wearing the HoloLens device and by giving his/her back to the manipulator;
- the real manipulator stands still in the fault configuration;
- as the alarm informs the user of a new fault, user can freely move around the environment to identify which type of fault has occurred and on which joint in the shortest possible time;

Once a fault is recognized, the same procedure is applied starting from the next position. Hence, each user has performed 9 different tests for each modality. To further avoid the learnability effects, the joints affected by the faults have been randomly selected out of the original number of manipulator's joints (6 for the Niryo robot) at each SP. Concerning the icons, only one fault could occur at a time and it has been randomly picked out from the original 3D icon set, discarding at each SP the icon displayed before. Hence, all the possible 3D icons have been evaluated in each system. Fig. 13 shows the test environment and the 9 SPs; arrows denote the user's starting orientation.

[10]https://bit.ly/2rJdF3M

**TABLE 5.** The questionnaire used to evaluate the subjective parameters.

| Category | Question | Score |
|---|---|---|
| QR1 | How many times have you used an AR application ? | 0 - 4 |
| | How many times have you used an Head-mounted Display for an AR application ? | 0 - 4 |
| | Have you ever worked with a robotic arm ? | 0 - 1 |
| | Do you know the Niryo robot ? | 0 - 1 |
| QR2 | It was easy to understand which type of fault has occurred. | 0 - 4 |
| | I could access the information at the most appropriate time and place using this specific AR device. | 0 - 4 |
| | The icons expressed the most correct amount of information | 0 - 4 |
| QR3 | The icon was positioned where I could see it. | 0 - 4 |
| | The icon was not positioned where I could see it. | 0 - 4 |
| | The icon had the correct dimension. | 0 - 4 |
| | The icon had not the correct dimension. | 0 - 4 |
| | The icon had the correct orientation. | 0 - 4 |
| | The icon had not the correct orientation. | 0 - 4 |
| QR4 | The FoV of the adopted device was suitable for the proposed use case. | 0 - 4 |
| QR5 | I rate this system as | 0 - 4 |

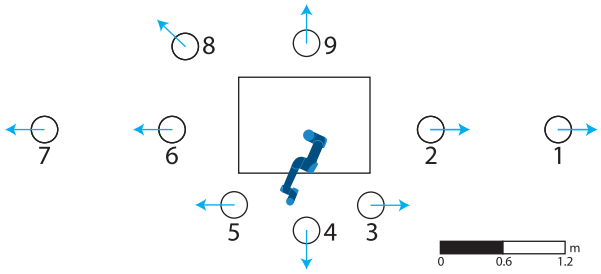



**FIGURE 13.** The test environment. The circles represent the SPs and the arrows the user's starting orientations.

Before starting the experiments, users have been introduced to the tests, showing them the 3D icons, along with their meaning, and the Niryo manipulator. Both objective and subjective parameters have been evaluated for the two modalities. Specifically, the objective parameters are the following:

- time required by users to recognize the fault and the related joint at each SP;
- number of errors in recognizing faults and joints at each SP;
- number of head translations and rotations required to recognize the fault at each SP.

To identify the virtual icons, users could either describe the icons or saying their names. Robot joints could be recognized by saying their numbers (from 0 to 5) or by pinpointing to them. Since the AR interface does not provide an interaction paradigm, users had to say by voice which fault has occurred and the number of the related joint to an external operator. In order to evaluate the subjective parameters, a questionnaire has been proposed to the users every time they completed the evaluation of a specific modality. The questionnaire was composed by 15 sentences, divided into 5 different categories. Before starting the evaluation of the two different modalities, the first four sentences (the QR1 category) regarding the user's general information, familiarity with AR and robotic arms have been proposed to the users. Two statements were ranked with a 5-point Likert scale (0 = never and 4 = every day), the others were Yes-No answers (Yes counted as 1 and No as 0). Then, after having completed the evaluation of a specific modality, users had to complete the remainder of the questionnaire, composed by 11 statements ranked in 5-point Likert scale (0 = strongly disagree and 4 = strongly agree):

1) QR2: clarity of the 3D icons. Whether the icons meaning was comprehensible by users;

**TABLE 6.** The objective results table.

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

**TABLE 7.** The subjective results table. For the sake of clarity, results have been normalized in the 0 - 100 interval.

| USERS | QR1 | | | | QR2 CLARITY | | QR3 PERCEPTION | | QR4 FOV | | QR5 SCORE | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | AR | HMD | ROBOT | NIRYO | A | NA | A | NA | A | NA | A | NA |
| 1 | 50 | 25 | 0 | 0 | 75 | 91.66 | 62.5 | 75 | 75 | 100 | 75 | 75 |
| 2 | 75 | 75 | 100 | 0 | 91.66 | 58.33 | 91.66 | 33.33 | 75 | 25 | 100 | 50 |
| 3 | 50 | 75 | 100 | 100 | 91.66 | 75 | 91.66 | 95.83 | 50 | 75 | 75 | 75 |
| 4 | 25 | 25 | 0 | 0 | 100 | 66.66 | 100 | 75 | 75 | 75 | 100 | 75 |
| 5 | 0 | 0 | 0 | 0 | 91.66 | 58.33 | 100 | 70.83 | 100 | 50 | 100 | 75 |
| 6 | 50 | 25 | 0 | 0 | 100 | 91.66 | 91.66 | 58.33 | 75 | 75 | 75 | 75 |
| 7 | 100 | 100 | 0 | 0 | 91.66 | 83.33 | 91.66 | 75 | 75 | 75 | 75 | 50 |
| 8 | 100 | 100 | 0 | 0 | 100 | 100 | 100 | 87.5 | 100 | 75 | 100 | 100 |
| 9 | 50 | 25 | 100 | 0 | 100 | 83.33 | 95.83 | 66.66 | 75 | 50 | 100 | 75 |
| 10 | 25 | 25 | 0 | 0 | 75 | 75 | 95.83 | 70.83 | 75 | 25 | 100 | 75 |
| 11 | 100 | 100 | 0 | 0 | 91.66 | 41.66 | 100 | 25 | 75 | 0 | 75 | 50 |
| 12 | 25 | 0 | 0 | 0 | 100 | 91.66 | 100 | 62.5 | 100 | 75 | 100 | 75 |
| 13 | 0 | 0 | 0 | 0 | 91.66 | 100 | 95.83 | 75 | 75 | 50 | 75 | 75 |
| 14 | 50 | 25 | 0 | 0 | 83.33 | 83.33 | 100 | 50 | 100 | 75 | 75 | 50 |
| 15 | 25 | 0 | 0 | 0 | 100 | 33.33 | 95.83 | 8.33 | 100 | 25 | 100 | 25 |
| 16 | 25 | 25 | 0 | 100 | 75 | 58.33 | 91.66 | 41.66 | 100 | 75 | 75 | 75 |
| 17 | 25 | 25 | 0 | 0 | 83.33 | 58.33 | 95.83 | 58.33 | 100 | 50 | 100 | 50 |
| 18 | 75 | 75 | 0 | 0 | 100 | 91.66 | 91.66 | 75 | 100 | 75 | 100 | 75 |
| 19 | 100 | 75 | 100 | 0 | 83.33 | 66.66 | 91.66 | 33.33 | 75 | 50 | 75 | 25 |
| 20 | 0 | 0 | 0 | 0 | 75 | 75 | 66.66 | 58.33 | 100 | 75 | 75 | 75 |
| 21 | 0 | 0 | 0 | 0 | 91.66 | 58.33 | 75 | 50 | 75 | 50 | 75 | 50 |
| 22 | 25 | 25 | 0 | 0 | 91.66 | 91.66 | 95.83 | 75 | 100 | 75 | 100 | 75 |
| 23 | 25 | 25 | 0 | 0 | 83.33 | 66.66 | 66.66 | 58.33 | 75 | 50 | 75 | 50 |
| 24 | 25 | 25 | 100 | 0 | 75 | 50 | 83.33 | 66.66 | 75 | 50 | 75 | 50 |
| 25 | 100 | 100 | 100 | 0 | 83.33 | 83.33 | 66.66 | 41.66 | 75 | 75 | 75 | 50 |
| 26 | 0 | 0 | 0 | 0 | 100 | 58.33 | 100 | 50 | 100 | 50 | 100 | 75 |
| 27 | 50 | 25 | 0 | 0 | 100 | 66.66 | 87.5 | 41.66 | 75 | 25 | 100 | 75 |
| 28 | 0 | 0 | 0 | 0 | 100 | 25 | 100 | 41.66 | 100 | 25 | 100 | 25 |
| 29 | 75 | 25 | 0 | 0 | 83.33 | 41.66 | 75 | 16.66 | 100 | 50 | 75 | 25 |
| 30 | 25 | 50 | 100 | 0 | 83.33 | 75 | 100 | 83.33 | 100 | 75 | 100 | 75 |
| 31 | 25 | 25 | 0 | 0 | 91.66 | 16.66 | 79.16 | 16.66 | 100 | 0 | 75 | 50 |
| 32 | 25 | 0 | 100 | 0 | 100 | 100 | 91.66 | 87.5 | 100 | 75 | 100 | 75 |
| 33 | 25 | 25 | 0 | 0 | 83.33 | 50 | 91.66 | 25 | 100 | 25 | 100 | 25 |
| 34 | 25 | 25 | 100 | 0 | 100 | 75 | 62.5 | 66.66 | 100 | 50 | 100 | 75 |
| AVG | 40.44 | 33.82 | 26.47 | 5.88 | 90.19 | 68.86 | 88.96 | 56.36 | 87.5 | 54.41 | 88.23 | 61.02 |
| STD | 32.57 | 33.07 | 44.78 | 23.88 | 9.05 | 21.54 | 12.13 | 22.40 | 14.10 | 24.20 | 12.66 | 19.64 |

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

Table 5 shows the employed questionnaire used to assess the subjective parameters. In the following section, the collected results are presented and discussed.

## VII. RESULT ANALYSIS

In this section, the objective and subjective results are introduced and examined.

### A. OBJECTIVE RESULTS

Objective results are related to the time, errors and number of movements required to identify a particular fault.

Concerning time and movements, three different datasets have been collected by the experimental tests, corresponding to the time, translations and rotations values. Since the 3D icons and the related joints have been randomly picked out at each SP, no correlation exists between a value obtained in $SP_i(A)$ (a starting position $i$ used in the A modality) and one obtained in $SP_i(NA)$, with $i = 1, ..., 9$. Hence, in order to determine a global score for each dataset, the values of each SP have been summed up for every user. Then, the average score of a dataset has been calculated dividing the sum of the total values by the number of users. Table 6 shows the time, translations and rotations values. In order to statistically analyze the collected data, a two-tailed t-test ($p = 0.05$) with unequal variance has been performed on the average values. The A modality has allowed users to recognize the typology of the fault and the related joint faster than the NA modality ($p = 0.0017$) and with a fewer number of translations ($p < 0.001$) and head rotations ($p < 0.001$), reducing the physical effort. Since the virtual icons were positioned close to the joint affected by the fault, users have been able to recognize both the fault's typology and the joint's number at the same time, without needing to change frequently their location and point of view. Moreover, the automatic scaling and orienting mechanism has allowed to keep the icon in the narrow FoV of the HoloLens device, maintaining it clearly recognizable. On the other hand, users have been forced to

change their point of view and position more frequently in the NA modality when the virtual icon could not be seen in the same way from several directions (e.g., both the velocity and break icons present a circular shape and they can be mismatched if not viewed from the front side). Finally, no errors have occurred during the recognition of the icons and the related joints. Since icons have been always uniquely and clearly identified, their design seems to be adequate for the proposed scenario. Moreover, the correct recognition of the joints suggests that the adopted tracking modality has been deemed suitable to correctly align the augmented assets to the real manipulator.

### B. SUBJECTIVE RESULTS

Similarly to the pre-process of the quantitative data, QR1, QR2 and QR3 results have been aggregated, summing the scores of the related questions for each user and calculating the average values (notice that, for the sake of clarity, the values have been mapped in the 0 - 100 range). Since QR4 and QR5 were made up by only one statement each, the aggregation has not been necessary. Table 7 illustrates the subjective results. Regarding QR1, users stated to have occasionally used AR applications and HMDs. The 26% has already worked with a robotic arm and only the 6% knew the Niryo Robot. A two-tailed t-test ($p = 0.05$) with unequal variance has been used to analyze the QR2, QR3, QR4 and QR5 results. Concerning QR2 and QR3, users have been able to better understand the icons meaning ($p < 0.001$) and they have deemed more suitable the position, scale and orientation values ($p < 0.001$) with the A modality. Since users may not have been able of entirely visualize the icons in the NA modality (the icon may not have fitted in the FoV), they may have found some difficulties in perceiving their intrinsic meaning (hypothesis that seems to be confirmed also by the time spent to recognize the virtual icons). Moreover, taking into account the HoloLens position, orientation and FoV, the A modality has allowed to determine position, scale and orientation values better suited for the considered use case. Hypothesis that seems to be confirmed also by the QR4 results ($p < 0.001$) which shows how the versatility of the A modality has allowed to reduce the unpleasant effects of the HoloLens narrow FoV. Finally, despite the NA modality has been deemed acceptable, the A modality has been preferred for the proposed use case in QR5 ($p < 0.001$).

## VIII. CONCLUSION

This paper proposes an adaptive AR system to display industrial robot faults using a wearable AR device. Starting from the identification of the most common industrial errors, a methodological approach has been employed to figure out which 3D virtual metaphors best evoke faults on industrial robots. Then, an adaptive modality to display the virtual metaphors has been presented. The user's position/orientation and the parameters of the device's FoV have been dynamically employed to position the virtual metaphors in areas close to the fault's location, always visible from the user, without occluding the robot itself.

In order to assess the effectiveness of the proposed solution, a comparison between the adaptive modality and a non adaptive one has been performed. Both objective and

**Pseudocode 2** The Pseudocode of the A Modality

**Input**:
pos, orient // *position and orientation of the user*
error_location // *Joint affected by fault position*
joint_values // *Joint orientations*
fault_type
**Constant**:
$r_{min}$, $r_{max}$, $D_{min}$, $D_{max}$, $D_{robot}$, $S_c$, $S_{min}$, $S_{max}$
**Start A1**:
camera = setCamera(pos, orient)
V = getCameraPosition(camera)
J = getPosJointFault(error_location, joint_values)
$D_{JV}$ = getJointCameraDistance( J, V)
$S_{icon}$ = null
**if** $D_{JV} < D_{min}$
  $S_{icon} = S_{min}$
**else if** $D_{JV} >= D_{max}$
  $S_{icon} = S_{max}$
**else**
  $S_{icon}$ = getScaleIcon($S_c$, $D_{robot}$, $D_{JV}$) // *see Eq. 3*
**end if**
**Start A2**:
icon = getIcon(fault_type) // *instantiate the 3D icon*
$BB_{3D}$ = get3DBB(icon)
$BB_{2D}$ = get2DBB($BB_{3D}$) // *on the camera plane*
$I_{hsv}$ = getImage(camera)
$I_t$ = getThresholdImage(camera, $r_{min}$, $r_{max}$)
$Q_s$ = getQuadrantsGrid($I_t$, $BB_{2D}$)
$Q_{start}$ = getStartingQuad($Q_s$)
$Q_{r_max}$, $Q_{c_max}$ = getRowColumnQMax($Q_{start}$, $Q_s$)
$EL_{max}$ = max($Q_{r_max}$, $Q_{c_max}$)
found_quad = false
**while**( found_quad == false):
  Q = getQuad($Q_{start}$, $EL_{max}$, k, u) // *see Eq. 6 and 7*
  **if** Q is FREE
    $Q_{selected}$ = Q
    found_quad = true
  **end if**
**end while**
ray = getRayThroughQ($Q_{selected}$)
R = getPointonRay(ray)
VR = getVector(V, R)
VJ = getVector(V, J)
$pos_{3D}$ = projection(VJ, VR) // *see Fig. 12*
**Start A3**:
look_vector = getLookAtVector(V, $pos_{3D}$)
getIconOrientationFrame(look_vector, icon) // *change icon's orientation*

subjective parameters have been evaluated. Results suggest that with the adaptive modality users have been able to recognize faults faster and with less movements than with the non adaptive solution. The capability of placing the icons in positions always visible from the users has allowed to reduce the troublesome limitations of the narrow FoV of the HoloLens device.

Future developments will be focused on testing the adaptive modality on the HoloLens itself evaluating the impact of the computational effort. Finally, to better understand the effectiveness of the adaptive modality, some tests will be carried out with high-payload industrial robots.

### A. PSEUDOCODE OF THE A MODALITY

The pseudocode of the A modality used to dynamically position the virtual metaphors.

## REFERENCES

[1] S. Isić, "Modernization and automation of automotive industry production processes with industrial robots," *Godina*, vol. 12, nos. 3–4, p. 105, Jul./Dec. 2015.

[2] I. Karabegović and E. Husak, "China as a leading country in the world in automation of automotive industry manufacturing processes," in *Proc. 4th Int. Congr. Motor Vehicles Motors*, Oct. 2016, pp. 06–08.

[3] I. Karabegoviá, "The role of industrial robots in the development of automotive industry in China," *Int. J. Eng. Works*, vol. 3, no. 12, pp. 92–97, Dec. 2016.

[4] A. Bauer, D. Wollherr, and M. Buss, "Human–robot collaboration: A survey," *Int. J. Humanoid Robot.*, vol. 5, no. 1, pp. 47–66, Mar. 2008.

[5] S. A. Green, M. Billinghurst, X. Chen, and J. G. Chase, "Human-robot collaboration: A literature review and augmented reality approach in design," *Int. J. Adv. Robot. Syst.*, vol. 5, no. 1, p. 1, Mar. 2008.

[6] I. E. Sutherland, "A head-mounted three dimensional display," in *Proc. Fall Joint Comput. Conf.*, Dec. 1968, pp. 757–764.

[7] P. Milgram and F. Kishino, "A taxonomy of mixed reality visual displays," *IEICE Trans. Inf. Syst.*, vol. 77, no. 12, pp. 1321–1329, Dec. 1994.

[8] F. De Pace, F. Manuri, and A. Sanna, "Augmented reality in industry 4.0," *Amer. J. Comput. Sci. Inf. Technol.*, vol. 6, no. 1, pp. 1–7, 2018.

[9] Y. S. Pai, H. J. Yap, and R. Singh, "Augmented reality–based programming, planning and simulation of a robotic work cell," *Proc. Inst. Mech. Eng., B, J. Eng. Manuf.*, vol. 229, no. 6, pp. 1029–1045, 2015.

[10] E. Ruffaldi, F. Brizzi, F. Tecchia, and S. Bacinelli, "Third point of view augmented reality for robot intentions visualization," in *Proc. Int. Conf. Augmented Reality, Virtual Reality Comput. Graph.* Cham, Switzerland: Springer, 2016, pp. 471–478.

[11] C. Vogel, C. Walter, and N. Elkmann, "Safeguarding and supporting future human-robot cooperative manufacturing processes by a projection- and camera-based technology," *Procedia Manuf.*, vol. 11, pp. 39–46, Jan. 2017.

[12] S.-K. Ong, J. W. S. Chong, and A. Y. C. Nee, "Methodologies for immersive robot programming in an augmented reality environment," in *Proc. 4th Int. Conf. Comput. Graph. Interact. Techn. Australasia Southeast Asia*, Nov. 2006, pp. 237–244.

[13] J. W. S. Chong, S. K. Ong, A. Y. C. Nee, and K. Youcef-Youmi, "Robot programming using augmented reality: An interactive method for planning collision-free paths," *Robot. Comput.-Integr. Manuf.*, vol. 25, no. 3, pp. 689–701, Jun. 2009.

[14] H. Fang, S. K. Ong, and A. Y.-C. Nee, "Robot programming using augmented reality," in *Proc. Int. Conf. CyberWorlds*, Sep. 2009, pp. 13–20.

[15] S. K. Ong, J. W. S. Chong, and A. Y. C. Nee, "A novel AR-based robot programming and path planning methodology," *Robot. Comput.-Integr. Manuf.*, vol. 26, no. 3, pp. 240–249, Jun. 2010.

[16] H. C. Fang, S. K. Ong, and A. Y. C. Nee, "Interactive robot trajectory planning and simulation using augmented reality," *Robot. Comput.-Integr. Manuf.*, vol. 28, no. 2, pp. 227–237, Apr. 2012.

[17] H. C. Fang, S. K. Ong, and A. Y. C. Nee, "Robot path and end-effector orientation planning using augmented reality," *Procedia CIRP*, vol. 3, pp. 191–196, Jan. 2012.

[18] H. C. Fang, S. K. Ong, and A. Y. C. Nee, "A novel augmented reality-based interface for robot path planning," *Int. J. Interact. Des. Manuf.*, vol. 8, no. 1, pp. 33–42, Feb. 2014.

[19] H. C. Fang, S. K. Ong, and A. Y. C. Nee, "Novel AR-based interface for human-robot interaction and visualization," *Adv. Manuf.*, vol. 2, no. 4, pp. 275–288, Dec. 2014.

[20] M. F. Zaeh and W. Vogl, "Interactive laser-projection for programming industrial robots," in *Proc. IEEE/ACM Int. Symp. Mixed Augmented Reality*, Oct. 2006, pp. 125–128.

[21] C. P. Quintero, S. Li, M. K. X. Y. Pan, W. P. Chan, H. F. M. Van der Loos, and E. Croft, "Robot programming through augmented trajectories in augmented reality," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2018, pp. 1838–1844.

[22] J. A. Frank, M. Moorhead, and V. Kapila, "Realizing mixed-reality environments with tablets for intuitive human-robot collaboration for object manipulation tasks," in *Proc. 25th IEEE Int. Symp. Robot Hum. Interact. Commun. (RO-MAN)*, Aug. 2016, pp. 302–307.

[23] Y. H. Su, C. F. Liao, C. H. Ko, S. L. Cheng, and K. Y. Young, "An AR-based manipulation system for industrial robots," in *Proc. 11th Asian Control Conf. (ASCC)*, Dec. 2017, pp. 1282–1285.

[24] M. Gradmann, E. M. Orendt, E. Schmidt, S. Schweizer, and D. Henrich, "Augmented reality robot operation interface with Google tango," in *Proc. 50th Int. Symp. Robot.*, Jun. 2018, pp. 1–8.

[25] D. Krupke, F. Steinicke, P. Lubos, Y. Jonetzko, M. Görner, and J. Zhang, "Comparison of multimodal heading and pointing gestures for co-located mixed reality human-robot interaction," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2018, pp. 1–9.

[26] H. Peng, J. Briggs, C.-Y. Wang, K. Guo, J. Kider, S. Mueller, P. Baudisch, and F. Guimbretière, "RoMA: Interactive fabrication with augmented reality and a robotic 3D printer," in *Proc. CHI Conf. Hum. Factors Comput. Syst.*, Apr. 2018, p. 579.

[27] A. Kron, G. Schmidt, B. Petzold, M. I. Zah, P. Hinterseer, and E. Steinbach, "Disposal of explosive ordnances by use of a bimanual haptic telepresence system," in *Proc. IEEE Int. Conf. Robot. Automat.*, vol. 2, Apr./May 2004, pp. 1968–1973.

[28] R. Marin, P. J. Sanz, P. Nebot, and R. Wirz, "A multimodal interface to control a robot arm via the web: A case study on remote programming," *IEEE Trans. Ind. Electron.*, vol. 52, no. 6, pp. 1506–1520, Dec. 2005.

[29] T. Machino, Y. Nanjo, Y. Yanagihara, H. Kawata, S. Iwaki, and K.-I. Shimokura, "Robot-augmented communication: A remote-collaboration system based on a shared field of view in real space," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Aug. 2005, pp. 2203–2209.

[30] V. Le Ligeour, S. Otmane, and M. Mallem, "Augmented reality interface for free teleoperation," *IFAC Proc. Volumes*, vol. 38, no. 1, pp. 523–528, 2005.

[31] Y. Xiong, S. Li, and M. Xie, "Predictive display and interaction of telerobots based on augmented reality," *Robotica*, vol. 24, no. 4, pp. 447–453, Jul. 2006.

[32] M. A. Al-Mouhamed, O. Toker, and A. Iqbal, "A multithreaded distributed telerobotic framework," *IEEE ASME Trans. Mechatronics*, vol. 11, no. 5, pp. 558–566, Oct. 2006.

[33] V. Le Ligeour, S. Otmane, M. Mallem, and P. Richard, "Distributed software architecture for collaborative teleoperation based on networked mixed reality platforms," in *Proc. 2nd Int. Conf. Inf. Commun. Technol.*, vol. 2, Apr. 2006, pp. 3498–3503.

[34] Y. Harada, N. Nazir, Y. Shiote, and T. Ito, "Human-machine collaboration system for fine assembly process," in *Proc. SICE-ICASE Int. Joint Conf.*, Oct. 2006, pp. 5355–5360.

[35] D. Ni, A. W. W. Yew, S. K. Ong, and A. Y. C. Nee, "Haptic and visual augmented reality interface for programming welding robots," *Adv. Manuf.*, vol. 5, no. 3, pp. 191–198, Sep. 2017.

[36] F. Brizzi, L. Peppoloni, A. Graziano, E. Di Stefano, C. A. Avizzano, and E. Ruffaldi, "Effects of augmented reality on the performance of teleoperated industrial assembly tasks in a robotic embodiment," *IEEE Trans. Human-Mach. Syst.*, vol. 48, no. 2, pp. 197–206, Apr. 2018.

[37] T. Chakraborti, S. Sreedharan, A. Kulkarni, and S. Kambhampati, "Alternative modes of interaction in proximal human-in-the-loop operation of robots," 2017, *arXiv:1703.08930*. [Online]. Available: https://arxiv.org/abs/1703.08930

[38] C. Vogel, M. Fritzsche, and N. Elkmann, "Safe human-robot cooperation with high-payload robots in industrial applications," in *Proc. 11th ACM/IEEE Int. Conf. Hum.-Robot Interact. (HRI)*, Mar. 2016, pp. 529–530.

[39] C. Vogel and N. Elkmann, "Novel safety concept for safeguarding and supporting humans in human-robot shared workplaces with high-payload robots in industrial applications," in *Proc. ACM/IEEE Int. Conf. Hum.-Robot Interact.*, Mar. 2017, pp. 315–316.
[40] C. Vogel, M. Poggendorf, C. Walter, and N. Elkmann, "Towards safe physical human-robot collaboration: A projection-based safety system," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Sep. 2011, pp. 3355–3360.
[41] A. Hietanen, R. Halme, J. Latokartano, R. Pieters, M. Lanz, and J.-K. Kämäräinen, "Depth-sensor–projector safety model for human-robot collaboration," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS) Workshop Robot. Co-Workers 4.0*, Madrid, Spain, 2018, pp. 3–6.
[42] O. Danielsson, A. Syberfeldt, R. Brewster, and L. Wang, "Assessing instructions in augmented reality for human-robot collaborative assembly by using demonstrators," *Procedia CIRP*, vol. 63, pp. 89–94, Jan. 2017.
[43] Z. Materna, M. Kapinus, V. Beran, P. Smrž, and P. Zemčík, "Interactive spatial augmented reality in collaborative robot programming: User experience evaluation," in *Proc. 27th IEEE Int. Symp. Robot Hum. Interact. Commun. (RO-MAN)*, Aug. 2018, pp. 80–87.
[44] I. Malý, D. Sedlávček, and P. Leitão, "Augmented reality experiments with industrial robot in industry 4.0 environment," in *Proc. IEEE 14th Int. Conf. Ind. Inform. (INDIN)*, Jul. 2016, pp. 176–181.
[45] R. Palmarini, I. F. del Amo, G. Bertolino, G. Dini, J. A. Erkoyuncu, R. Roy, and M. Farnsworth, "Designing an AR interface to improve trust in human-robots collaboration," *Procedia CIRP*, vol. 70, no. 1, pp. 350–355, Jan. 2018.
[46] R. S. Andersen, O. Madsen, T. B. Moeslund, and H. B. Amor, "Projecting robot intentions into human environments," in *Proc. 25th IEEE Int. Symp. Robot Hum. Interact. Commun. (RO-MAN)*, Aug. 2016, pp. 294–301.
[47] C. Fantuzzi, C. Secchi, and A. Visioli, "On the fault detection and isolation of industrial robot manipulators," *IFAC Proc. Volumes*, vol. 36, no. 17, pp. 399–404, Sep. 2003.
[48] V. D. Singh and V. Banga, "Overloading failures in robot manipulators," in *Proc. Int. Conf. Trends Elect., Electron. Power Eng. (ICTEEP)*, 2012, pp. 15–16.
[49] J. Chen and R. J. Patton, *Robust Model-Based Fault Diagnosis for Dynamic Systems*, vol. 3. New York, NY, USA: Springer, 2012.
[50] A. Dietrich, M. Schulze, S. Zug, and J. Kaiser, "Visualization of robot's awareness and perception," in *Proc. 1st Int. Workshop Digit. Eng.*, Jun. 2010, pp. 38–44.
[51] F. De Pace, F. Manuri, A. Sanna, and D. Zappia, "An augmented interface to display industrial robot faults," in *Proc. Int. Conf. Augmented Reality, Virtual Reality Comput. Graph.* Cham, Switzerland: Springer, 2018, pp. 403–421.
[52] D. De Tommaso, S. Calinon, and D. G. Caldwell, "A tangible interface for transferring skills," *Int. J. Social Robot.*, vol. 4, no. 4, pp. 397–408, Nov. 2012.
[53] L. Claassen, S. Aden, J. Gaa, J. Kotlarski, and T. Ortmaier, "Intuitive robot control with a projected touch interface," in *Proc. Int. Conf. Social Robot.*, 2014, pp. 95–104.
[54] Z. Zivkovic, "Improved adaptive Gaussian mixture model for background subtraction," in *Proc. 17th Int. Conf. Pattern Recognit.*, Aug. 2004, pp. 28–31.
[55] F. Lamberti, F. Manuri, G. Paravati, G. Piumatti, and A. Sanna, "Using semantics to automatically generate speech interfaces for wearable virtual and augmented reality applications," *IEEE Trans. Human-Mach. Syst.*, vol. 47, no. 1, pp. 152–164, Feb. 2016.
[56] J. Keil, F. Schmitt, T. Engelke, H. Graf, and M. Olbrich, "Augmented reality views: Discussing the utility of visual elements by mediation means in industrial ar from a design perspective," in *Proc. Int. Conf. Virtual, Augmented Mixed Reality*, 2018, pp. 298–312.
[57] H. Lin, W. Lin, W.-C. Tsai, Y.-C. Hsieh, and F.-G. Wu, "How different presentation modes of graphical icons affect viewers' first fixation and attention," in *Proc. Int. Conf. Universal Access Hum.-Comput. Interact.* Cham, Switzerland: Springer, 2015, pp. 226–237.
[58] H. Lin, Y.-C. Hsieh, and W. Lin, "A preliminary study on how the icon composition and background of graphical icons affect users' preference levels," in *Proc. Int. Conf. Hum. Aspects IT Aged Population*. Cham, Switzerland: Springer, 2016, pp. 360–370.
[59] H. Lin, Y.-C. Hsieh, and F.-G. Wu, "A study on the relationships between different presentation modes of graphical icons and users' attention," *Comput. Hum. Behav.*, vol. 63, pp. 218–228, Oct. 2016.
[60] W. K. Horton, *The Icon Book: Visual Symbols for Computer Systems and Documentation*. Hoboken, NJ, USA: Wiley, 1994.



**GIANCARLO AVALLE** received the B.D. and M.D. degrees in cinema and media engineering from the Politecnico di Torino, Turin, Italy, in 2016 and 2019, respectively.



**FRANCESCO DE PACE** received the M.Sc. degree in cinema and media engineering from the Politecnico di Torino, Italy, in 2016, where he is currently pursuing the Ph.D. degree with the Dipartimento di Automatica e Informatica, under Prof. Andrea Sanna. From 2016 to 2017, he was a Mobile Developer with Consoft Sistemi, Italy. His research interests include human–computer/human–robot interaction, augmented reality, and virtual reality.



**CLAUDIO FORNARO** received the M.Sc. degree in electronic engineering and the Ph.D. degree in computer engineering from the Politecnico di Torino, Torino, Italy, in 1994 and 1998, respectively. He is currently a Researcher with UniNettuno University, Rome, Italy. He has coauthored several articles in various areas, such as embedded system (for astrophysics instruments), mathematics (special functions), education, computer security, and computer graphics.



**FEDERICO MANURI** received the B.Sc., M.Sc., and Ph.D. degrees in computer engineering from the Politecnico di Torino, Italy, in 2008, 2011, and 2017, respectively, where he is currently a Postdoctoral Research Assistant with the Dipartimento di Automatica e Informatica. His research interests include human–machine interaction, computer graphics, and augmented reality.



**ANDREA SANNA** received the M.Sc. degree in electronic engineering and the Ph.D. degree in computer engineering from the Politecnico di Torino, Turin, Italy, in 1993 and 1997, respectively, where he is currently an Associate Professor with the Dipartimento di Automatica e Informatica. He has authored or coauthored several articles in the areas of computer graphics, virtual and augmented reality, distributed computing, and computational geometry. He is a Senior Member of the Association for Computing Machinery.