Received: 10 July 2023 | Accepted: 19 February 2024

DOI: 10.1111/mice.13185

INDUSTRIAL APPLICATION

COMPUTER-AIDED CIVIL AND INFRASTRUCTURE ENGINEERING WILEY

Augmented reality-based method for road maintenance operators in human–robot collaborative interventions

A. C. Bavelos | **E. Anastasiou** | **N. Dimitropoulos** | **G. Oikonomou** | **S. Makris**

Laboratory for Manufacturing Systems and Automation, Department of Mechanical Engineering and Aeronautics, University of Patras, Patras, Greece

Correspondence
S. Makris, Laboratory for Manufacturing Systems and Automation, Department of Mechanical Engineering and Aeronautics, University of Patras, Patras 26504, Greece.
Email: makris@lms.mech.upatras.gr

Funding information
European Commission Research; H2020 Societal Challenges, Grant/Award Number: 955269

Abstract

Road maintenance operators often work in dangerous environments and are in need of a support system to enhance their safety and efficiency. Augmented reality (AR) has proven to be useful in providing support to operators in various industrial sectors. However, the vast majority of the existing applications focus mainly on static, controlled environments, such as industrial shopfloors, although the dynamic flow of information that AR can provide could be very valuable to unstructured, dynamic environments. This paper presents a novel AR—based methodology for human–robot collaboration, real-time instructions, and support for road maintenance operations, aiming to enhance operator safety and efficiency. A robot operating system-based architecture is exploited for the communication of the modules. The methodology is tested in a laboratory environment, and the results validate the hypothesized enhancement of road operators. In the future, the application will be validated on real interventions in highways.

1 | INTRODUCTION

Research on augmented reality (AR) has spread across many sectors, such as healthcare, education, and marketing, as well as manufacturing, which provides an ideal example of the enhancement AR can have on operators. Manufacturing in the EU is changing from fixed, non-dynamic systems that favor high-volume and low-variety productions, toward more flexible and reconfigurable systems, that favor variety in production, focusing on increasing the utilization of resources and reducing scrap rate (Chryssolouris, 2006; Makris, 2020). AR is one of the enabling technologies for the transition of the old-fashioned operator to a flexible one. It can be deduced that the characteristics that make this technology valuable for manufacturing systems can also be used for other sectors, and to enhance road operations in collaboration with technologies such as robotics, specifically the flexibility and reconfigurability in road operations.

Road maintenance is taxing work when done manually, when road interventions like barrier removal and placement must take place. Aside from the ergonomic point of view, the safety of the operator is also compromised, especially in high-traffic highways that are dangerous to the operator, who is standing still in comparison to the running traffic. Several papers research traffic-management methods, such as Jiang and Adeli (2004), Karim and Adeli (2003), and Ghosh-Dastidar and Adeli (2006) who researched traffic management methods using an object-oriented model, a case-based reasoning model, and a mesoscopic-wavelet model, respectively. The speed of the cars on the highway makes most accidents involving

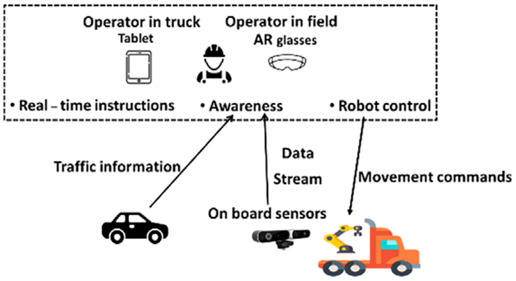


FIGURE 1 Augmented reality (AR) for road maintenance.

workers fatal. Statistics show that the road worker death toll in the EU in 2020 was 3355 people, showing an increase from the year before by 53 people, according to Eurostat. An obvious solution for these accidents is to reduce the number of road workers on the road, by automatizing the processes. Unfortunately, the process of road maintenance is done in a dynamic environment with dynamic parameters, and as such, it is not possible yet to complete all maintenance interventions without the interference of the human factor. As such, another solution to complement the first is to minimize the time that the road workers will stay on the road executing the maintenance processes or assist them in a way that they do not have to get out of the truck that they use as transportation and as a platform to execute the interventions.

Therefore, the solution of a mobile robotic shell in road maintenance, enhanced by auxiliary tools based on AR would greatly enhance safety. The robot can assist with maintenance interventions, taking the place of some operators. This concept has been gaining focus in recent research. In Katsamenis et al. (2022), the authors proposed a robotic vehicle that is supported by autonomous drones to coordinate maintenance works, while Eskandari Torbaghan et al. (2020) propose an automated robotic system for road crack sealing by using 3D printing techniques.

While automated processes are usually more efficient in repetitive and heavy-load actions than manual processes performed by humans, and processes that require decision making are more efficient when a human performs them, it is the collaboration between humans and robots that allows for the efficient execution of a larger and more diverse number of processes, which often require the human operator's intellect and decision making combined with the robot's advantages. AR is a great enabler of human–robot collaboration (HRC). The concept that will be presented in this paper is shown in Figure 1. The operator or operators will make use of a mobile robot platform, which comprises a truck with a mounted robot on the rear, to perform road interventions. The operator will use a tablet when they are in the truck, and it is not necessary to be on the road for the intervention, and while when they are on the road, they will use AR glasses. In both cases, they will be able to communicate with the robot, receive visualization from sensors and receive instructions as well as traffic information for road awareness. There are several challenges in this approach regarding AR, owing mostly to the dynamic nature of the road environment, especially regarding lighting that may change rapidly, the possible differences in road layout, changing traffic conditions, and the location of the road assets in relevance to the robot, which is not static.

This study's main objective is to present the advantages of a suite of tools consisting of AR-related technologies to support road maintenance workers, by enhancing their efficiency and safety in road interventions, through robot control, instructions, and real-time information feedback. The novelty, and computational originality of the proposed method and the advancement beyond the previously mentioned studies in AR operator support, is the development of a tailored solution for HRC in road interventions, integrating AR, leveraging a mobile robot platform, and addressing challenges specific to dynamic road environments. These unique elements collectively contribute to the advancement of the field and offer a promising approach for efficient and effective road interventions. The method is also built to be easily adaptable to any road intervention or any kind of process where the operator can be supported by the use of AR. It also proposes a method of enhancing the awareness of the operator in a very wide and uncontrolled area, something that is also not prevalent in the state of the art. The organization of the paper is as follows: Section 2 presents the literature review, Section 3 presents the proposed methodology of the solution, and Section 4 reviews the implementation of the discussed AR applications. In Section 5, experiments and a road-intervention case study are presented where the methodology is validated. The last section is dedicated to conclusions and future work.

2 | LITERATURE REVIEW

2.1 | HRC

A lot of previous research has focused on AR as a means for indirect HRC. HRC is an important element of the factories of the future, with safety being one of the most important aspects as pointed out in Maurtua et al. (2017), where a method for safety in HRC is proposed based on the robot's perception. AR can enhance these methods by also providing the human with enhanced perception as suggested by this paper. AR, either through holographic displays such as the ones provided by AR glasses or through video approaches such as through a tablet with camera feedback, can be used for robot control when the operator is in the area of the robot. Ni et al. (2017) proposed a method for remote manipulation of a robot combined with haptic feedback through an AR display, although the operator's view seems limited due to the use of one camera angle in the display. In Liu et al. (2018), the authors propose a non-traditional method for robot programming using an AR interface to visualize a decision-making process, supervising the robot's operation and showcasing great interoperability between AR and other state-of-the-art technologies. Other intuitive uses of enhanced graphics include Zhu and Veloso (2016) where a method was proposed for the visual enhancement of videos that showcase mobile robot movements, in order to extract the movement algorithms, although with the drawback that processing is done after recording. In Dimitropoulos, Togias, Zacharaki, et al. (2021), the operator can move the robot to the endpoint they desire by moving a virtual end-effector superimposed over the real robot's end-effector. The robot operating system (ROS) is often used in collaboration with AR, for example, in Lotsaris, Gkournelos, et al. (2021) and Lotsaris, Fousekis, et al. (2021), to control the robot without the user having prior experience in robotics. In both Lotsaris, Gkournelos, et al. (2021) and Lotsaris, Fousekis, et al. (2021), the AR application is connected to ROS, enabling robot control by connecting a virtual robot with the real one. Robotic control has also been implemented through AR glasses in Kyjanek et al. (2019) where a construction worker would be able to plan trajectories and view diagnostic feedback of the process. Again, ROS is used, and the operator can visualize plans created by ROS when given the target assembly and execute the plan that they are satisfied with. Additional tools can also be used with AR glasses instead of hand gestures. Projectors are an often-used tool as in Lee et al. (2016) where a camera unit and a projector are used in collaboration with a robot in an unconventional manner with loose kinematic specifications, although projectors would not work in dynamic environments. In Ong et al. (2020), the user can move around a workplace that includes a robot and use a handheld pointer to define the paths they want the robot to follow. In Hietanen et al. (2020), an AR GUI was proposed in combination with a monitoring system based on depth sensors, for safe human–robot-collaboration. Lambrecht et al. (2021) discuss the bottlenecks that prevent the widespread use of AR in industries, marking the spatial referencing of machines and AR devices as an issue, which is also discussed in later sections, as the problem is even worse in uncontrolled environments such as the highway. The above papers prove that AR is a great enabler in HRC. The ones that were considered the most in this paper for their ease of use and accuracy were the end-effector-based movements.

2.2 | Instructions and visualizations

Except human–robot interaction, the most usual utilization of AR in the industry is for the support of the operator with real-time instructions and information feedback on the status of the shopfloor. This is useful for both new and inexperienced operators as well as experienced operators who have to work on a new product, thus enabling high-variety productions. Additionally, AR serves to keep the human in the loop by connecting them with the digitalized machines in the shopfloor, enabling communication between them, as AR is a very effective interface for the communication of humans with machine intelligence (Baroroh et al., 2021). In Pai et al. (2016), the authors proposed a framework that uses AR for teaching and learning factory operations, although it focuses on immersion in their simulations, for which virtual reality may be more fitting. AR instructions have been prevalent in assembly processes in the last decade (Rentzos et al., 2013). In Makris et al. (2013), the authors proposed a concept for the generation of assembly instructions, which was then visualized in AR for the benefit of the operator. The proposed framework does not assist only operators in the shopfloor but also production engineers who generate the instructions without excessive burden. The authors in Eiriksdottir and Catrambone (2011) recognized three types of instructions regarding abstraction, instructions that explain each step of the task, instructions that provide information about rules and regularities, and demonstrations of a specific instance of the task that is carried out. In Michalos et al. (2016), the AR application does not only support the operator with instructions on the assembly but also keeps them aware of the state of the production and connects them with the robot on the shopfloor. Dimitropoulos, Togias, Michalos & Makris, 2021 proposed a method for support of the operator during assembly, with the instructions consisting of text and 3D models, combining more than one AR medium, AR glasses, smartwatch, and projector, enhancing AR visualization of instructions with classification of assets through the use of a neural network. Wang et al. (2016) introduced an interactive AR assembly guidance system that offered different modes of guidance for assembly operators during different stages of cognitive processing. An interesting application was developed by Si-Mohammed et al. (2018), where AR was combined with brain–computer interfaces (BCIs) to test the approach of guiding the design of UIs based on BCIs in AR. There is a truly sizeable number of publications on AR guidance from the past two decades, which proves the usefulness of the technology in operator support. This paper aims to build upon existing knowledge on the use of AR as a support tool in manufacturing and transfer the results in the road maintenance field, where they will prove just as valuable.

2.3 | AR and robotics in road maintenance

Robotics and AR have been gaining traction in road maintenance in the last few years. Karelina et al. (2022) prove that robots are an excellent tool in road construction, by proposing an autonomous robotic manipulator for the control of a bulldozer during highway construction. In Sheta and Mokhtar (2022), the authors propose an autonomous robot system for pavement inspection, constituting of a robot equipped with an Android phone that gathers a stream that is used to recognize the presence of cracks using convolutional neural networks. Inspection seems to be one of the prime targets of robotics in road maintenance. Also, in Shim et al. (2023), the proposed method consists of an inspection robot on a ground vehicle, able to be autonomously driven in tunnels. Tarek and Marzouk (2021) propose a handheld mobile AR application for the visualization of all inspection data using building information modeling (BIM) technologies during infrastructure maintenance activities. Ren et al. (2022) and Malek et al. (2023) propose a method for a real-time crack detection system with an AR interface, in real-time, and with no need for an external processing device. The authors in Sadhu et al. (2023) utilize in their research both AR and VR in combination with BIM technologies for structural health monitoring.

The research in this field makes apparent the enhancing effect that extended reality applications and robotics can have in maintenance operations.

2.4 | Gap to be addressed

The gap in the state of the art that this paper aims to bridge is the lack of a method of AR operator support that works for dynamic and dangerous environments with no fences and in adverse conditions outside the traditional factory environments, such as a highway. The method that will bridge this gap must take into account both efficiency and most importantly the safety of the operators. In the next sections, an AR-based methodology is presented to fulfil these criteria.

3 | METHODOLOGY

In order to enable safety and efficiency in road maintenance operations, a methodology for the deployment of a suite of AR technology tools is proposed as shown in Figure 1

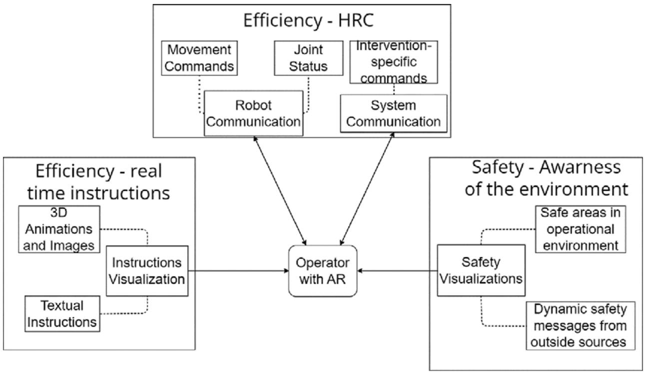


FIGURE 2 AR methodology block diagram. HRC, human–robot collaboration.

A block diagram showcasing the proposed methodology is shown in Figure 2.

The arrows in the image show the flow of information, and the dashed lines show the relationship between a main module—the one that is directly connected to the operator—and the functionalities that it consists of. The diagram shows three distinct modules that are combined to form the method.

The safety block shows the relationship between safety and awareness of the dangerous environment of a dynamic and dangerous work environment. As is the norm with AR, the update of information that aims to make the operator aware is done through visualizations. These visualizations consist of safe areas in the immediate environment of the operator as well as dynamic updates on events that happen in the wider range of the operator, depending on the type of environment. For example, in highway environments, the operator should have an awareness of a range spanning some kilometers close to them. Aside from events, the operator can also receive feedback from sensors, such as video streams or images of specific angles in the environment that they cannot see directly.

The real-time instructions that are attributed to efficiency in the proposed method consist of visualization of animations of 3D models, images, and text. The arrow shows that the flow of information goes to the operator, but that is not to say the communication is one way, as the operator could also, for example, affect the interface to go to the next instruction when they desire. The arrow shows the important flow as it regards the wider methodology. Communication, as with the previous block, is done through the AR interfaces, depending on the AR device that the operator uses. Efficiency also indirectly affects safety. When the operator is more efficient, they spend less time performing the task in a dangerous environment, lessening the danger to their person.

The last block shows human–robot-collaboration (HRC), which is also considered to enhance efficiency. The communication here is done both with the AR interfaces and with the communication framework between AR and the robotic system, which will be analyzed later in the paper. Communication with the operator and this block is a two-way street since the operator must be aware of the exact status of the robot in the shopfloor, and the robot must receive movement commands from the operator. Aside from pure HRC, this method also supports communication with the general system of the operation and the transference of information and commands to other entities on the shopfloor, such as a machine that the operator can command to begin executing its functionality.

The methodological approach to the functionality of the blocks shown in Figure 2 will be analyzed further in this section. In each sub-section, an example will be given to show how the functionalities are implemented in two road maintenance interventions, the "safety barrier installation" and "crack sealing" interventions, depending on the functionality, if it is more useful for the AR glasses or the tablet. In the safety barrier installation, the operator using the AR glasses must handle a robot that is mounted on a truck that contains new barriers and with the help of the robot install a new barrier in place of a faulty one on the highway. The robot is the one carrying the barrier, while the operator must screw the barrier on the pillars as the robot holds it. In the crack sealing case, the operator is inside the truck and must use the tablet to handle the robot that has a crack sealing machine as a tool in order to seal cracks on the highway.

3.1 | AR instructions visualization

AR instructions are provided to the operator while they are in the process of executing their tasks. They are meant to sufficiently explain the steps of their task, as well as the way they are supposed to be taken, and this is achieved in the form of text, images, videos, or 3D models. Text is used in usually short but accurate descriptions of the tasks, with helpful instructions on how the operator may execute them.

Additional visual input is also needed, and it is one functionality in which AR applications and headsets excel at. 3D digital models can be overlayed in the real world at positions where the real objects must be placed, drilled, or screwed in. The operator can watch the highlighted objects, search for the real ones in the equipment of the assembly, and simulate the movement and operation that the animated 3D models display. 3D models of the tool to be used, such as a screwdriver's model, inform the operator of which tool they are supposed to use in each specific step. In order for the instructions to be as seamless as possible and not be an obstacle for the operator to complete their tasks, the 3D models disappear when the operator closes in their vicinity, to not visually inhibit them, using the hand tracking of the AR glasses and calculating the minimum distance from the 3D visualizations.

Aside from 3D models—computer-aided design (CAD) models, which are created with 3D design tools such as Blender—helpful visualizations also include images. Images can play a similar role to 3D models, in cases where steps are more easily displayed in an image. Other visualizations can also play a part in instructions. A driver in the truck who must reach the place where the intervention takes place, in which the robot performs actions, must have guidelines. The guidlines are superimposed over camera feedback from a camera mounted on the back of the truck, and are displayed in the tablet application, showing the optimal and maximum range of the robot to help the driver take a proper place while driving in reverse.

In the proposed method, the instructions consist of all the above examples. For each step of the interventions where the glasses are used, an accompanied text appears on the instructions panel that floats in the view of the operator. In case it is applicable, 3D models are spawned overlayed on the real world, to assist the operator with the process. For example, in the safety barrier installation, where the new barrier must be screwed in the pillars, virtual 3D screws are superimposed over the holes they must be inserted in and a text instructing the operator that it is time to screw in the barrier is shown in the panel. In case a tool is needed for the completion of the task, such as a screwdriver, it is also visualized. In case the tablet is used, as is the case of the *crack sealing intervention*, the instructions have some differences according to the particularities of the medium. The operator in this case is inside the truck, and they receive feed from the outside world through cameras mounted on the back of the truck. Text is still used to explain every step to the operator, such as an instruction to approve or modify a crack detection, and visualizations are overlayed over the feed from the cameras, such as the lines of the detected cracks, colored to show the exact positions of the detection output. Figure 3 presents the method followed for the real-time instructions, where the operator receives all the types of information explained above. Safety zones are only shown in the AR glasses since the operator is outside the truck and works closely with the robot, while they are unneeded when the operator is inside with the tablet. Animations and text work for both the AR glasses and tablet as they are the crux of the instructions, while the tablet makes great use of video feedback and images, which the AR glasses do not need due to their direct view of the environment.

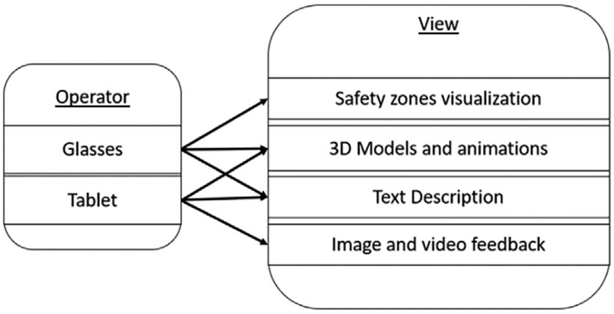


FIGURE 3 AR instructions visualizations.

3.2 | AR for HRC and system commands

Aside from being on the receiving end of information when using AR devices, the application enables the operator to send commands to connected devices or, most importantly, to the central control system of each operation. This serves a multitude of purposes, such as alerting the system of an operation start or end, intervening in the process if necessary, or commanding the resources of the operation with easy, user-friendly interfaces, bypassing the need to gain the expertise to handle the resources by their own interfaces.

The concept of this paper, which is to support the operator in their cooperation with a modular robotic platform, renders the manipulation of robots by the operator necessary. Usually, to program and manipulate a robot, the operator would have to use a teach pendant or program it from a computer that is connected to the same network as the robot. In cases where every motion is predefined, this type of programming is preferred since it is performed long before the operation takes place. However, to control the robot via an AR application, the ROS (Quigley et al., 2009) can be utilized. ROS is an open-source middleware framework designed to develop and control robots. It provides services and tools for hardware abstraction, communication between processes, package management, and more, facilitating the development of robotic software. When building the application, the robot model and its URDF—meaning the file that describes the robot's physical attributes to ROS—are imported into the AR application and loaded as a virtual robot. Utilizing specific libraries, which will be covered in the next section, the virtual robot and the real robot are connected, and the application receives the state of the real robot constantly, essentially its joint states. The way ROS communication is established is further analyzed in Section 4.2, in the Robot Control section. This communication goes both ways, allowing for the AR application to affect the robot. In this application, for the case of the AR glasses, a user-friendly gesture system was developed. The operator is able to move a virtual end-effector, placing it wherever they want the end-effector of the real robot to end up, the end-effector being a tool or a gripper in our cases. Then they can prompt the system to create a plan for this position. The control system sets the virtual end-effector position as the target and calculates the inverse kinematics with its planners. The trajectory output is received by the AR application and visualized in 3D in the glasses with the help of the virtual robot, called the "ghost robot," a semi-transparent model overlayed over the real one. The operator watches the trajectory, then, if they are satisfied, they confirm it, or they send for a re-plan. As soon as they confirm, the application sends the command to execute the trajectory in ROS, and the real robot moves. For example, in the *safety barrier installation*, when the operator receives the instruction to move the barrier that the robot is grasping to the pillars, they grab the virtual end-effector, move it close to the pillars, and request a trajectory, which is presented to them via the ghost robot animation. If they are satisfied, they command the execution of the trajectory, and the barrier is brought close to the pillars.

With the tablet, where the operator is inside the truck, they see the robot from several views from on-board cameras. They can then move the robot via arrows, which they use to move the virtual end-effector, gaining the 3D perspective from the different views. Then, same as the glasses, they send for a plan and then confirm or ask for a re-plan. Aside from that, there is also the option of changing the joint values of the robot directly from the controller, bypassing the ROS planners. For example, in the *crack sealing*, the operator needs to bring the robot in a position to view the road and the cracks on the road with a camera that is mounted on the tool. The operator can see the view of several cameras from several angles that overlook the robot area, and with buttons, they move the virtual end-effector, which is superimposed over those streams; then they request a trajectory and watch it from several angles, and if they are satisfied, they order the execution of the trajectory, which brings the robot to the desired position.

Aside from robot control, the communication between AR and ROS allows the user to send other messages or call services in the ROS-based system. Operations can be called, such as the identification of a road asset from the on-board cameras, as in the crack detection in the *crack sealing intervention*, or the opening and closing of the gripper of the robot, as in the *safety barrier installation*, where the operator orders the gripper to release the screwed in the barrier in its final position. This allows the operator to be in control of the intervention and the resources that have digital interfaces. The communication aspect from a technical point of view is analyzed in the next section. As a concept, the operator is able to affect the system through interfaces in AR, panels, and buttons,

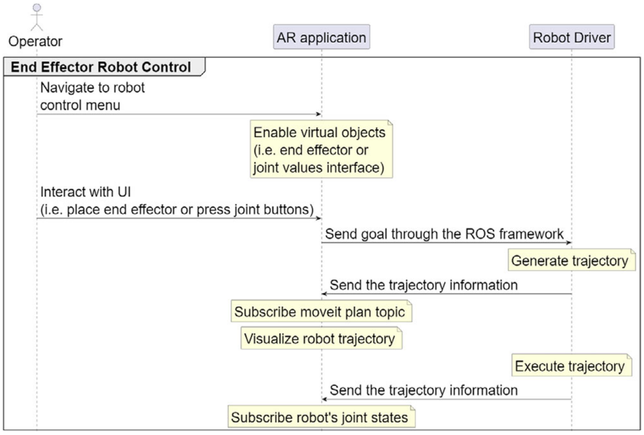


FIGURE 4 Sequence diagram for robot control. ROS, robot operating system.

as well as voice commands, commanding pre-existing functions in ROS, in the form of services or actions.

The logic of the robot control process is shown in the sequence diagram of Figure 4.

AR's primary role in the interventions is to keep the operator in the loop and in control. It is one of the goals of the application to provide full situational awareness to the operator. When the operator is in the field, wearing the AR glasses, they can view the intervention space with their own eyes. The AR application enhances their experience by providing monitoring of the execution, since the instructions showcase the steps that the operation is at any given point, and also provides information about the safety zones and information about the surrounding environment, completing the operator's situational awareness since they are aware of both the operation and their surroundings. In the case where the operator is inside the truck, the operator has no view of the outside environment. In this case, the tablet application provides the view from video streams of outside cameras, and also provides monitoring of the operation, and again provides information on the wider surroundings. Therefore, in both cases, the operator has full situational awareness and is kept in the loop about the operation, the robot, and the overall system. The more seamless the application, the greater the efficiency and the reduction of the cycle time of operations. The architecture of the graphical user interface (GUI) that is created to implement the above functionalities is shown in Figure 5, essentially summarizing the operator's interactions with the AR application during intervention runtime.

The panel consists of a canvas with fields where the operator can fill with text using a virtual keyboard that is spawned when they click on the text fields. Once they input their credentials in these fields, the application presents the second panel while the connection with the server scene is established. The server scene is a second scene of the same application that runs on the glasses or tablet, but this scene runs on a connected PC. The server scene is responsible for outside communication, such as the communication with the robotic system, and transfers the necessary data to the scene deployed in the AR device. The second panel is the configuration of the application. In this phase, the operator must input any parameters that are needed for the accurate function of the application in the intervention. That includes the localization of the scene and the robot, a process which redefines the points of origin of the animations in the application. It also sets them in the appropriate place for each intervention and will be analyzed in more detail in the latter parts of this section. After the animations are configured, the operator moves on to the main menu. The main menu contains the choice between intervention and robot control. If the robot control button is pressed, the application moves to a robot control mode, in which the operator can use gestures to move the robot. In case the interventions button is pressed, a choice of interventions is presented to the operator, and by choosing one, they can choose to begin the process and the instructions. Each instruction is presented in a panel that contains the text that guides the operator, while 3D animations are included where appropriate.

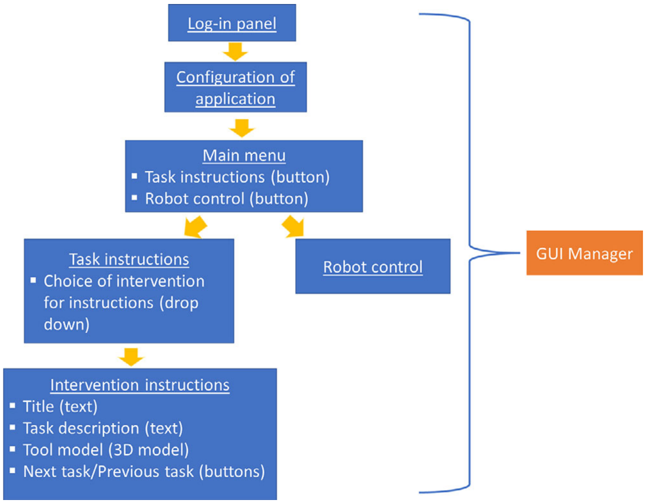
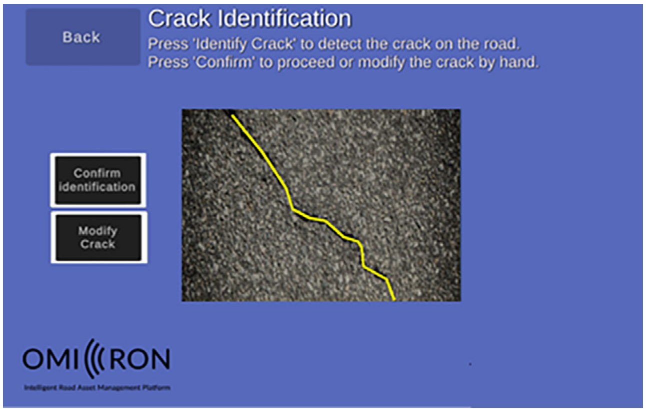


FIGURE 5 UI menu architecture. GUI, graphical user interface.

3.2.1 | Localization

In order to move the robot accurately, and for the animations to appear in the correct spots in the real world, a correlation must be established between the real and virtual worlds through localization, using markers and virtual objects. This also compensates for differences in road layouts. Before any operation is executed, two calibrations must be performed. The first calibration pertains to the robot and any other virtual objects that are placed or moved in relation to the robot's position (i.e., the virtual end-effector). However, the distance between the robot's position and the position of the other real-world objects—safety barriers (from the intervention "safety barrier installation") will be used as an example—may vary. The robot will be loaded onto a truck, and the driver will stop the truck near the activity area, which may consist of real objects on the road, such as safety barriers or road signals. Despite the driver's experience, it is impossible to stop the truck, each time, at the exact same distance of the activity area. In addition, the areas of the same interventions differ due to road layout differences. Therefore, the first calibration alone is not sufficient for positioning animations and virtual objects relative to the activity area. The second calibration is used to place all animations and virtual objects in the correct positions relative to the real objects of the activity area, and it also enables the calculation of specific robot movements.

Before localization, all the superimposed visuals are rendered according to an initial reference system defined by the operator's position in the real world the moment the application starts. During the localization process, this reference system is translated and rotated along with any virtual objects, to a new position that will be defined by the calibrating object. This object can be either a marker or a virtual object that the operator can place using an application's interaction method. After tracking the marker or position of the virtual object, the operator inputs the initial pose of the localization into the AR application and executes the calibration function. The initial pose refers to the offset and rotation between the calibrating object and the starting point of the desired reference system for the visuals to be rendered at the correct positions. The AR application detects the position of the calibrating object, applies the initial pose, and sets the reference system's starting point at the calculated position. After the calibration process, all visuals are rendered according to their offset and rotation from the new reference system's starting point.



Press 'Identify Crack' to detect the crack on the road.
Press 'Confirm' to proceed or modify the crack by hand.

FIGURE 6 Corrected trajectory in tablet.

3.2.2 | Trajectory refinement

Other ways of controlling the robot are also possible by using AR interfaces. One example is seen in the case of a "crack sealing" intervention, in which the robot will follow a specific trajectory to seal the crack, with a crack sealing tool. The system must detect cracks on the road and calculate a trajectory for the robot to follow in sealing the crack. The operator initiates the process by pressing a button in the AR application's UI, which triggers a service in the system to perform the detection. This system is based on computer vision-based techniques, but the details are not the subject of this paper, which is focused on the AR visualization of the output. Once the detection is complete, the trajectory is calculated and sent to the AR application as a series of 2D points relative to the camera's viewport. The application then creates a line connecting these points and displays it on the tablet's UI. The operator can view the feedback from the camera superimposed with this line. If the detection is successful, the line should be rendered precisely on top of the crack as seen in Figure 6. If any errors are detected in the crack's identification, the operator can modify the trajectory. When the operator presses the button to modify the crack detection, the previous line is erased, and a function is activated to allow the operator to draw the new trajectory using the tablet's touch screen. The detection might contain more than one crack, and they can be erased and re-added separately. The function collects points for each 2 mm of drawing in the camera's viewport. After the new line is drawn, the operator can choose to modify it again or confirm the result. If the result is confirmed, a function is called, which performs a homographic transformation. After this transformation, the resulting points are transferred to the system for calculation of the robot's motion.

To calculate the final trajectory of the robot, the collected points need to be transformed to account for the depth between the camera and the projection of each point on the road. Crack detection uses a camera on the robot's gripper at a specific height from the ground. The trajectory is displayed in a 2D canvas within the AR application, and the modified points from the operator are in a 2D coordinate system. To calculate each point in the 3D coordinate system, a homography transformation is then performed according to the theory below.

In Figure 7, C represents the known position of the camera. There are three different height levels depicted:

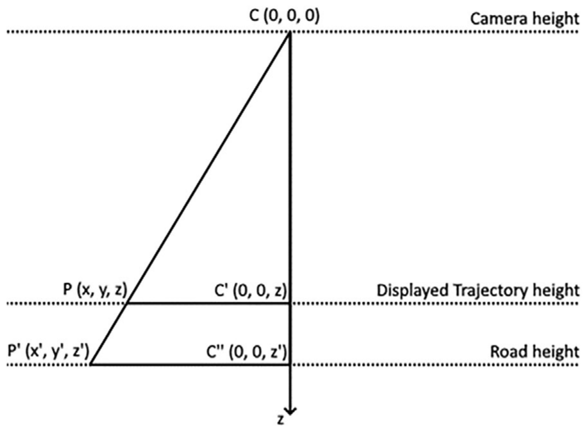


FIGURE 7 Camera view to road correlation.

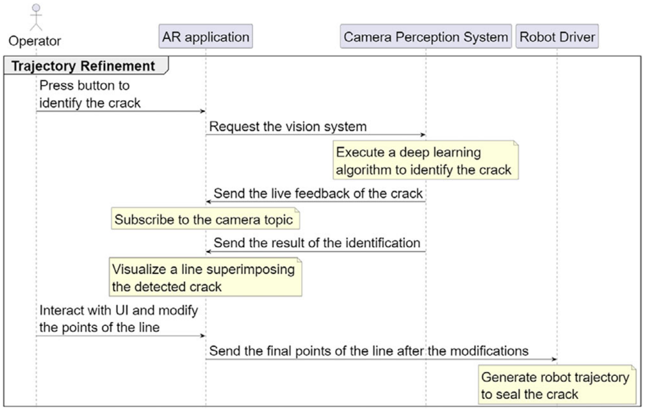


FIGURE 8 Sequence diagram for trajectory refinement.

the height of the camera (set to zero), the height of the displayed trajectory (the vertical distance in the AR application between the camera and the virtual line representing the crack), and the height of the road (the actual height of the camera from the ground level in the real world). Points C′ and C″ are the two perpendicular projections of point C in the line and road height levels, respectively. Point P represents one of the collected points while the operator was modifying the crack in the tablet's UI, and point P′ is the actual point that the system will use to calculate the final trajectory of the robot's movement.

The point P′ is calculated as follows. The two right-angle triangles of Figure 7 (CC′P and CC″P′) are similar. According to the theory of the similar triangles, the following formula is used:

$$\frac{CC'}{CP} = \frac{CC''}{CP''} \tag{1}$$

or,

$$\frac{z}{\sqrt{x^2 + y^2 + z^2}} = \frac{z'}{\sqrt{{x'}^2 + {y'}^2 + {z'}^2}} \tag{2}$$

Solving for x' and y':

$$x' = x * \frac{z'}{z} \tag{3}$$

and

$$y' = y * \frac{z'}{z} \tag{4}$$

After the transformation, the calculated points can be used to generate the final trajectory of the robot's movement. The system receives the collected and transformed points, calculates the final trajectory, and sends it to the AR application for display. The operator can view a hologram of the robot executing the trajectory and then initiate the motion. The crack modification process is shown in Figure 8.

3.3 | AR for information feedback and safety

Aside from the instructions that are analyzed above, the operator can receive information from many sources that can be visualized in several ways, as AR excels in information visualization as presented by the authors in Siatras et al. (2021). This enables the operator to be fully aware of their environment. When the operator is in the truck with the tablet on hand, such as in the case of crack sealing, the tablet, as mentioned above, can receive image feedback from any of the on-board cameras that have a view of the road and the operation and are connected via ROS, IP, or a similar protocol to the tablet. The views are from different angles to have a 3D perspective of the world, overseeing the most important part of the intervention, essentially the place where the tools will be utilized.

Important feedback that the operator receives can be information about the road, such as traffic information, or information about accidents close to the intervention site. The purpose of this is to enhance the safety of the operators on the road, by making them as aware as possible of the situation on the highway. This enables them to take actions to protect themselves, such as leaving the scene. Other useful information for the intervention can also be monitored from the AR devices such as temperature, which is important for some interventions such as crack sealing, as well as changes in the weather such as incoming rain or strong winds. This information can be gathered from any server that is regularly updated regarding highway information. Although the point of this paper is not to show the server side, but rather what can be achieved by the communication, in this case, the application receives data from V2X communications (MacHardy et al., 2018). Additionally, the application generates visualizations for safety, most importantly safety zones. When the operator is working close to a machine or a robot, they are in danger if those assets are not programmed to work with humans, such as non-collaborative robots. For the visualization of the safety zones, the application receives real-time data through the ROS connection that regards the geometry of the zones. These data include points in the 3D space that are used by the application to generate virtual objects and display the zones with the correct geometry and dimensions. According to the specific use case, these data can be published by safety scanners, cameras, or any other device that is included in the integrated safety system. This way the operator can stay at a safe distance from heavy machinery. Other safety visualizations could also be alerts when the operator gets close to the limits of the safety lane they are working in. In the safety barrier installation, the operator works outside the truck, close to the robot, and also in combination with it during the screwing where the robot holds the barrier and the operator screws in the bolts. During the time that the robot is moving, the operator sees the zones indicating the reach of the robot and stays at a safe distance.

4 | IMPLEMENTATION

This section aims to explain the implementation plan that will validate the methods proposed in the previous section. For the development of AR applications, the framework that serves as the base is Unity3D, a development platform that is primarily used for game development, as well as mixed reality applications. Unity3D has a graphical interface to create and manipulate objects and also a scripting API based on C#, with which scripts with various functions can be created and attached to the objects to define their behavior. For communication with the robot, the ROS was used, an open-source middleware that comprises a set of s/w libraries that is primarily used to build robotic applications. In Figure 9, the main architecture for the ROS-Unity and Client-Server communication is shown. The different modules in the diagram will be explained in this section. The separation between ROS and Unity is shown with the Unity parts being inside the box. They are connected via the ROSbridge protocol (Crick et al., 2017). The ROS architecture and the relevant ROS topics are explained in more detail in the Robot Control section. The network communication describes the communication between two scenes, the server scene, which runs in a PC server called broker, and the client scene, which runs in the AR device and are detailed in the GUI section. The Unity visualizations have been analyzed in the previous section. The CAD models refer to the 3D models that the application uses for the animations.

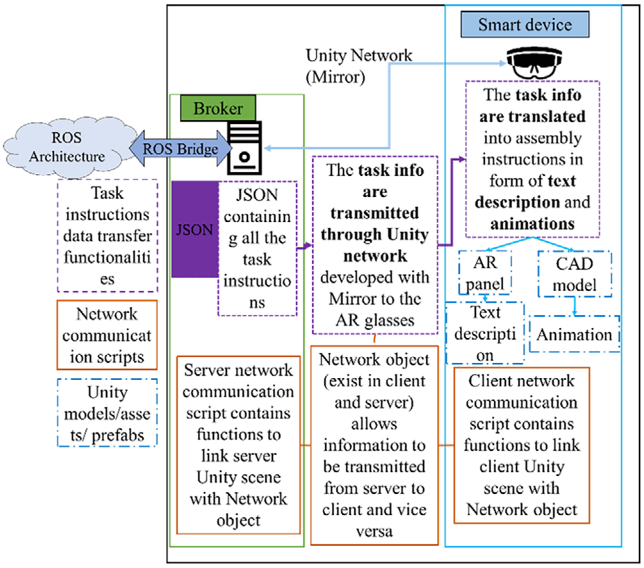


FIGURE 9 Robot operating system (ROS)-Unity and client–server communication. CAD, computer-aided design.

4.1 | AR instructions and control commands

4.1.1 | GUI

For the provision of instructions, a UI menu is created for the tablet and the AR glasses, which serves as a panel for the visualization of text instructions and as the primary interface for the interaction of the operator with the robotic system. A network of communication is created between two scenes, as they are called in Unity3D that are akin to applications. The first one is called the client scene, which is the scene that is deployed in the AR device. The second one is the server scene, which is run on a PC on the same network as the AR device.

Communication between the two scenes is achieved using Unity's MIRROR communication (Lindblom, 2020). The server scene contains the instructions, in the form of a JSON file. This JSON file contains the descriptions of the tasks, the tools that may be needed in each step, the name of the 3D models assigned to each step, and the axis of their movement in the animation, as well as their speed, meaning that holographic parts will move to the axis and the distance given from the JSON files, while text instructions will be shown on UI menu. Of course, the instructions must be shown to the client application, which is the one deployed in the AR devices. The client connects to the

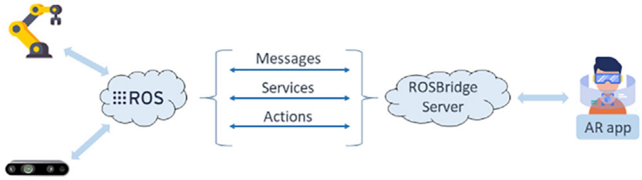


FIGURE 10 AR communication through ROS framework.

server and receives the instructions and animations from it, spawning the objects in its own interface.

The AR server also serves as a broker server, facilitating the connection between the AR application and the robotic system. By utilizing this server, the operator can transmit and receive information regarding the ongoing operation. If the server receives a message from the system, it initiates a Client Remote Procedure Call to transmit the necessary information to the client. Conversely, when the operator intends to control the robotic system manually, such as manipulating the robot's movement, the client scene sends a command to the AR server, which subsequently dispatches the appropriate message to the system.

Moreover, the AR server enables the transmission of information to multiple clients simultaneously. This functionality was employed in the AR application to establish a connection between two client scenes, one running on the AR glasses and the other on the tablet. As it will be explained in later section, there are instances where messages need to be transmitted from the system to both clients and from one client to another.

4.2 | Robot control

As mentioned, aside from instructions, the AR application also provides, to the operator, the control of the operation. As mentioned, the primary framework for communication is ROS.

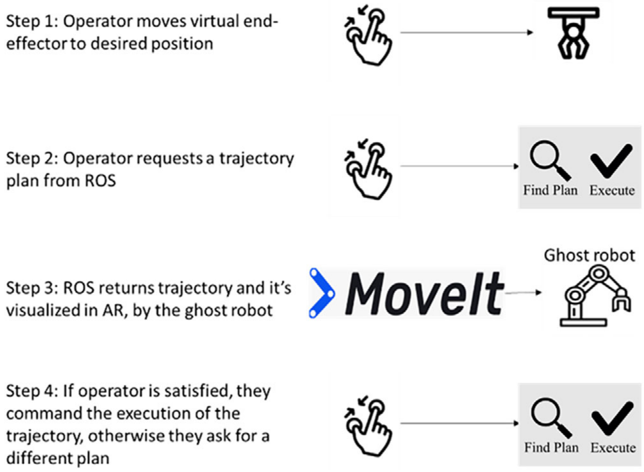
ROS primarily works through topics, which can be considered as the medium that transfers the messages created by ROS nodes, which include the functions of our method. Topics are a unidirectional form of information flow, where anyone who wants the information can subscribe and receive the messages. In case a different type of communication is needed, requiring a response to the request for information, ROS services are used. While Unity3D and its applications are not ROS-based, the connection is made via the ROS# library for Unity3D, in combination with ROSbridge protocol, which is run on Ubuntu in the same network as the ROS system and allows for the transfer of ROS messages through topics to non-ROS applications via Websocket transfer. A basic outline is shown in Figure 10. In this way, the AR application subscribes to the topics shown in Table 1.

TABLE 1 Robot operating system (ROS) topics.

ROS topic	Description
/joint_states	The augmented reality (AR) app receives the joint states of the robot from this topic
/manipulator_controller/command	The AR appl publishes to this topic to move the robot by changing the joint states, as an alternative method of moving than the cartesian
/retry_plan_topic	The AR app publishes to this topic when the operator wishes to replan and create a new trajectory of the robot
/execution_topic	The AR app publishes to this topic when the operator is satisfied with the plan and wishes to continue with the execution
	The AR app subscribes to this topic to visualize the plan that Moveit! makes
/move_group/display_planned_path	The AR app subscribes to this topic to visualize the safety zones
/safety_zones_topic	The AR app subscribes to this type of topic to receive images from ROS-connected sensors
/camera/color/image_raw/compressed	This is an example of a service topic that the AR app publishes to when the operator wants to send a specific command to ROS, such as the command to begin the identification of the sign. The names of these topics vary according to their function

Robot control is achieved through an action server/client architecture. According to this architecture, the capability for cartesian movement of the robot is encapsulated in a server handler, essentially a script that is running with the functions necessary for the cartesian movement of the robot and waiting for an action message to pass the parameters for the movement, such as the target endpoint of the movement, the speed, or the name of the robot. Through the ROS# library, action clients can be developed in the AR application, which sends the action message to the ROS action server, with the relevant parameters. The way this is done in the application is shown below in Figure 11.

The process shown in Figure 11 was simulated with AR and a virtual robot, a 3D model with fake ROS controllers, which publish the same topics as the real ones would. As shown in the image, and as mentioned in Section 2.2, the



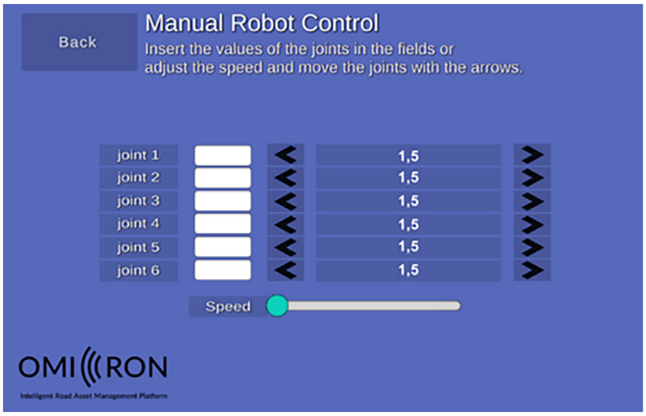
Step 1: Operator moves virtual end-effector to desired position

Step 2: Operator requests a trajectory plan from ROS

Step 3: ROS returns trajectory and it's visualized in AR, by the ghost robot

Step 4: If operator is satisfied, they command the execution of the trajectory, otherwise they ask for a different plan

FIGURE 11 Manual robot control with gestures.



Manual Robot Control

Insert the values of the joints in the fields or adjust the speed and move the joints with the arrows.

FIGURE 12 Robot control, tablet version.

TABLE 2 Robot and activity worlds.

Robot world	Activity world
Virtual robot	KUKA robot animation that places the barrier on the columns
Ghost robot (hologram)	Fastening screws animation
Safety zones	Final position of new barrier near the columns
Virtual end-effector	KUKA robot animation that places the barrier on the columns
Virtual robot	KUKA robot animation that places the barrier on the columns
Ghost robot (hologram)	Fastening screws animation

operator grabs the end-effector of the robot and moves it to the desired position and then presses the button to see the trajectory before the actual movement. This is enabled by the use of the "ghost" robot, a more transparent virtual 3D model of the robot in this simulation case, which displays the plans made by ROS Moveit! (Chitta, 2016). The operator checks the movement for collisions with dynamic objects, meaning objects that are not already in the planning scene of Moveit!, which is a scene that contains static obstacles in relevance to the robot, such as the truck. If an object already is in the planning scene of Moveit!, then it is taken into account in the generation of the trajectory. Then, if the operator agrees with the trajectory, they can press the execute button, which will allow the robot to continue with the actual movement. If they do not, for example, in case the ghost robot passes through another object, constituting a collision, the operator can order a new plan to be made until they are satisfied.

For the tablet, the procedure is mostly the same, the only difference being that with the tablet, the movement of the end-effector is done via arrow buttons since gestures cannot be used.

An additional menu for the robot control in the tablet is presented in Figure 12. In this case, the operator can see the joint states of the robot and can change them with arrows, with the capability to change the speed.

The two applications, for glasses and for tablets, need to be able to communicate between themselves since there are cases where they are used together. In one case, the driver of the truck of the mobile robotic platform may be getting information on the process by the tablet while the operator is outside and receiving complementary instructions. One example of this case is when the operator outside of the truck must move the robot manually. In this example, the driver who has access to the controller of the robot must press the free-drive button to enable the manual control by the other operator. The connection between the devices in this case is achieved using the MIRROR network as explained above.

4.2.1 | Localization

As mentioned in Section 3.2, before each operation is executed, there are two calibration processes that must be performed in order to establish a relation between the real and the virtual worlds. To perform these two calibrations, two different objects have been created in the application. The first object is called "Robot World," and it serves as the reference system for the robot and other virtual objects that are calibrated along with it. The second object is called "Activity World," and it sets the reference system for the area where the activity will be executed. Table 2 shows all the virtual objects and target positions for the robot's movements that are calibrated along with these two reference systems.

The first calibration requires the operator to use AR glasses to scan a quick response (QR) code, which is placed at a specific position near the robot's base. The tracking of the QR code is performed using the official Microsoft Mixed Reality Toolkit (MRTK) SDK for QR code tracking, developed by Microsoft and supported by the Hololens 2

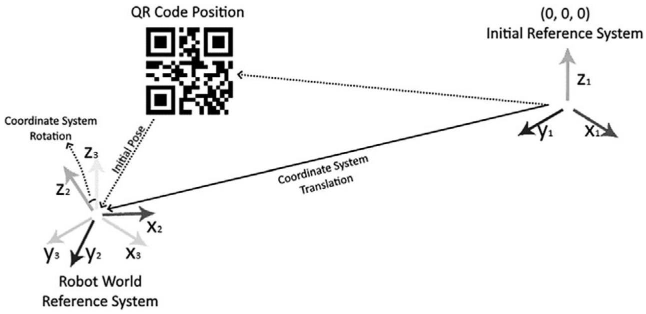


FIGURE 13 Initial reference system.

AR glasses. For optimal performance, the QR code should be either version 1–10 or a micro QR code M1-M4. The initial reference system is shown in Figure 13.

The implementation of localization with QR code markers is a quick, efficient, and accurate method for performing a calibration in this application. However, it requires that a QR code be printed and placed at a fixed position on the robotic cell, either on the truck or on a stationary part of the robot, such as the robot's base. This ensures that the marker moves with the robotic cell and is not situated in the surrounding environment, such as on the road. The area for installing a new barrier is not standard, unlike a typical work area. As a result, the second calibration requires the use of a virtual object to serve as a reference for the translation and rotation of the coordinates system, similar to the role played by the QR code in the first calibration.

Before starting an operation, the virtual object will be displayed in front of the operator wearing AR glasses in the application's viewport. This object has the appearance of the prime object of the intervention, in this paradigm, a virtual barrier, with the same size and shape as the real barriers on the side of the road. The operator can manipulate this virtual barrier and place it in the position where the new barrier (currently loaded on the truck) will be placed. Once the operator places the virtual barrier in the correct position, the same calibration function as before will be executed to perform the transformation of the "Activity World" object. Unlike the first calibration, there is no initial pose in this case. In the first calibration, the requested starting point of the coordinates system is the center of the robot's base, while in the second calibration, the starting point is set to the center of the virtual barrier, and thus both the initial offset and initial rotation are null vectors.

To perform the localization, a calibrating function is called in the AR application. This function executes all the necessary mathematical transformations to find the position of the requested coordinate system. After setting the reference object (QR code or virtual barrier), this calibrating function is called by the operator, either using the user's interface or activating the proper voice command. To begin the execution of the calibration, this function accesses the following inputs: the calculated distance between the start of the initial system and the reference object, the virtual objects that refer to the coordinate systems (Robot World and Activity World), and the initial pose, which consists of the initial offset and the initial rotation. The value of the initial pose is by default inserted by the developers in the AR application. However, if the operator is not satisfied with the calibration, a user interface has been developed to allow the adjustment of this value. The operator can change this value and perform the calibration again in order to achieve the optimal localization.

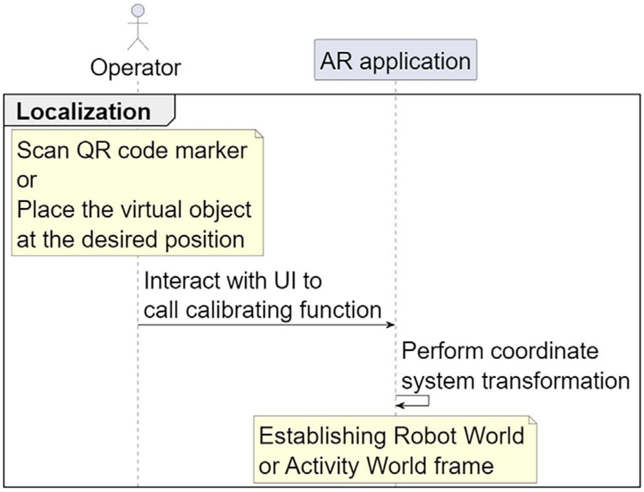


FIGURE 14 Sequence diagram for localization. QR, quick response.

One last detail it is worth mentioning about the localization is that in order to make the communication between the AR application and the robot (through the ROS), all the values that are inserted in the calibrating function are given according to the ROS coordinate convention. Before any calculation, these values are transformed in the coordinate convention of Unity3D by the calculating function. To perform this transformation, the function accesses two other functions. The first is called "Ros2Unity" and transforms the initial offset value. The second one is an official method in Unity, which is called "Quaternion.Euler" and performs the conversion from degrees to quaternion values. The logic of the localization process is shown in the sequence diagram of Figure 14.

4.3 | AR for information feedback and safety

As mentioned in Section 3.3, information feedback is performed with the use of sensors to relay the information. In the proposed method, the sensors are cameras. There are several ways to obtain feedback from each camera, depending on the type and manufacturer. Cameras for which there are ROS drivers can transmit their streams as topics. These topics may contain images, depth data such as pointcloud, camera parameters, and so forth. Visualization is done by accessing these topics from Unity with the use of ROS-bridge and the ROS# library. The topic that transfers the image message is shown in Table 1. Once these topics are subscribed to by Unity scripts, the information can be visualized by assigning it to a relevant Game Object. For example, one can add an image object in the scene in Unity and assign the image received by the image topic of the camera to visualize the image in Unity. This method is used in this application for multiple tasks, such as the remote manual control of the robot through the tablet's interface, the monitoring of the robot's movements through the execution of an operation, and the observation of the robot's future trajectory. This way of receiving camera info is common for all cameras with ROS drivers. The specific cameras used in the implementation of the method were Realsense and ZED cameras.

Other forms of information visualization can also be received through ROS, such as geometrical data, which can be used by Unity to overlay shapes, like the outline of a road signal, over the camera view. Information may also be given in the form of text on a panel. Unity contains scripts and objects for many kinds of visualization, depending on the libraries used. In the proposed method, MRTK is the principal library for visualization and interactions.

The way that V2X information can be received depends on the platform that is used to broadcast it. The platform itself is not the scope of this paper but rather the visualization of the information. In this case, an implementation was used as a simulation of V2X information, which can be received from AR via commands specific for Apache protocol. Essentially what is retrieved is a JSON, which contains in its fields the valuable information that the application needs. This JSON is parsed, and the field that contains the information is extracted according to its name, for example, "Accidents." This information was visualized in a panel, which keeps a history of all messages that the operator can browse.

4.4 | Lighting challenge

Visualization in AR glasses is not simple, as the lighting plays a very important role. When the environment is indoors, with controlled lighting, the virtual models and panels and in general, all the visualizations presented by the AR glasses can be seen clearly, especially if the brightness has been adjusted by the user. But, due to their transparency, the models cannot easily be seen in sunlight, bordering on the impossible. As mentioned in the Introduction, the road maintenance intervention environments are dynamic since they take place on the road, and while some of them are executed at night, some are done during the day. The lighting depends on the sky, and there is a high chance that the daylight will be bright. In such a case, a practical solution was found to solve the issue, specifically, the use of tinted film, such as the one used by car windows. This reduces the amount of light—both visible light and ultraviolet light rays—that penetrates the glasses. The design shown in Figure 15 will be used by the operator when daylight affects their vision in the glasses. It is easy to remove and put on quickly, which is useful when there is a rapid change in lighting.

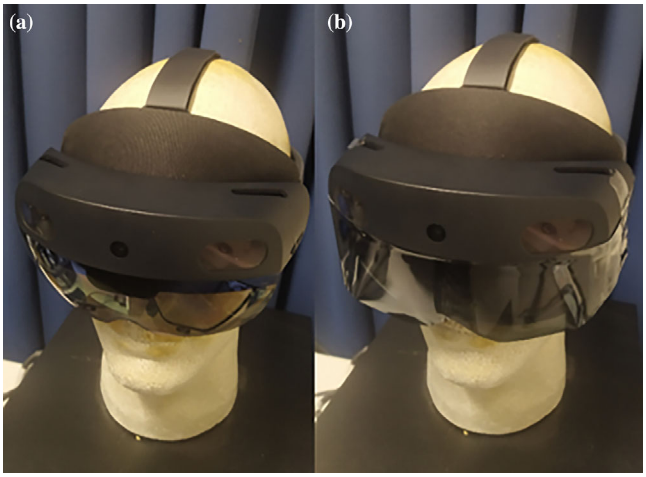


FIGURE 15 Light filter on AR glasses (a) Hololens 2 (without filter) and (b) Hololens 2 with solar film applied.

5 | EXPERIMENTS

The results and demonstrations that follow aim to prove the feasibility of the proposed method in road interventions, concerning all the facets of the AR application, in a lab-scale environment, and set the standard for the improved versions that will be validated on the road in future works. The proposed method will be used in several use cases or, more accurately, road interventions. The technologies described above are assigned to the different use cases according to their functions and how useful they are in each case. A table of interventions and their assigned technologies can be seen below in Table 3. The assignment was done with the following logic in mind. If the operator needs to do work, out of the truck, using their hands during the intervention, they wear the AR glasses since they need their hands free. Additionally, the glasses offer augmentation in the operator's actual view, which is also a safety

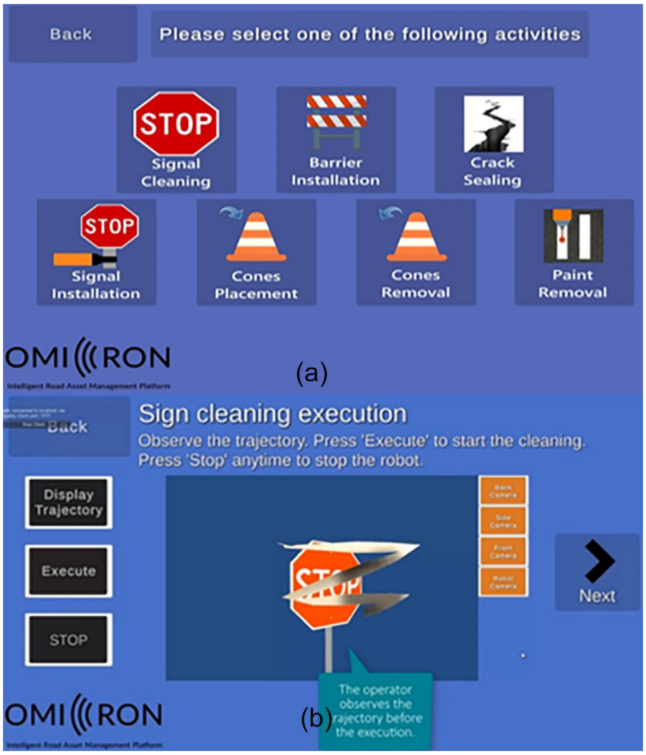
TABLE 3 Use cases assignment.

Technology Intervention	AR–tablet	AR–glasses
Cone placement/collection	X	
Signals placement		X
Signal Cleaning	X	
Laser-based paint removal	X	
Crack sealing	X	
Safety barriers		X

issue, since looking at a tablet while on the highway is dangerous. In case they stay in the truck, where their view is impeded and therefore the glasses are of less use, there is no safety issue and the tablet is better for watching camera streams, and the tablet is used. A brief explanation of how the AR applications will be used in the road interventions follows and then examples of the interfaces, with the robot and road assets in simulation. Last, an experiment based on the barrier installation is presented for a lab-scale validation of the proposed method, and conclusions are drawn in a qualitative analysis.

It is obvious from Table 3 that the most used application in the different use cases is the tablet one. In this case, as mentioned above, the operator is inside the truck and executes the operation from there. In cone placement and collection, the robot that is mounted on the truck places cones on the road or collects them on the back of the truck. The tablet application is used in cone placement and collection for the visualization of the rear view of the truck where the robot is grasping the loaded cones to deposit them on the road or collecting the cones that were previously on the road, and it is also used to allow the operator to command the robot to release the cones with a button. An example of the interface of the tablet application is shown in Figure 16.

In signal cleaning, the robot is equipped with a water hose and cleans signals that are detected by a camera mounted on the robot. The tablet application is used in this case for the visualization of the signal and its detection for approval by the operator, as well as the visualization of the robot trajectory plan also for approval, and for controlling the robot to move it to its pre-detection position. Several on-board cameras present the operation to the operator, who is inside the truck, from various angles. In laser-based paint removal, where the robot has a mounted laser as a tool and erases paint from the road, the tablet is used for robot control in the intervention where the operator needs to move the robot themselves. For the crack sealing intervention, where the robot has a tool that fills in cracks on the road, the tablet is used to display to the operator who is inside the truck the view of cameras that display the process that happens on the road. It is also used for guidelines visualization for the driver of the truck and for modification of the robot trajectories that the detection system—that the crack sealing system uses—calculates. It is evident that these use cases consist of common functionalities that are adjusted for each one accordingly but stem from the same basis. These functionalities are: the provision of instructions to the operator about each step of the interventions, the communication of the AR application with the central ROS system, sending commands such as “close gripper” or “begin process,” the feedback from cameras to provide the operator with a clear view of the working area from different cameras, the functionality of ordering and monitoring planned trajectories of the robot, as well as executing them, and the visualization of useful information about the intervention, such as the detection of cracks or signals, as well as the real-time update of environment updates from V2X.



Please select one of the following activities

Sign cleaning execution

Observe the trajectory. Press 'Execute' to start the cleaning. Press 'Stop' anytime to stop the robot.

The operator observes the trajectory before the execution.

FIGURE 16 Tablet UI interface: (a) choice of intervention and (b) display of robot trajectory using mounted camera on the robot.

The AR headset is used in the signal installation, in which the robot grasps signals stored in the back of the truck, using a specifically designed gripper for the geometry of the signals, and assists the operator, who is on the road, in installing them. In this case, the operator with the headset is able to call ROS services and receive safety information as well as real-time instructions. In safety barrier installation, in which the robot assists the operator in installing safety barriers on the road in places where old barriers need to be replaced, the AR application offers the operator the capability of robot control through gestures, allows them to call ROS services, and of course, provide safety and real-time instructions. For the glasses, the operator can use gestures to interact with the UI, but they also have the capability to interact and move through the UI easily through voice commands. The common functionalities that these use cases consist of are the instructions to the operator with text and 3D animations superimposed over the real environment, the visualization of safety zones, the functionality of ordering, and monitoring planned trajectories of the robot, as well as executing them, the commands to the ROS system, such as “close gripper” and the real-time update of environment updates from V2X.

As mentioned above, the applications have been tested standalone for each use case, with virtual assets, a virtual robot, and a virtual mobile platform, although most capabilities, like the robot control, have been tested with real equipment as well.

The virtual robot is a KUKA KR60, as this is the robot model that will be used when the interventions that are now simulated will be implemented in demonstrations on the road in the future. On the ROS side, the virtual robot’s controller is programmed to output the same topics as the real robot’s controller would. As such, on the AR side, the actions that were developed to send commands to the robot’s topics will not be any different when the virtual robot is exchanged for the real one, making the future integration much easier and proving the method to be adaptable to any hardware system. The simulations were used to test the interfaces and the instructions, regarding ease of use and comprehension, respectively, and remodeled according to the results.

To demonstrate in more practical terms the application, a physical setup was created that simulated the barrier installation environment, in-lab, as shown in Figure 17a. The robot that was used was a UR10e, with a suction gripper made with three suction cups, to grip a barrier that was made from plexiglass to simulate the real barrier. The UR10 can only handle a load of 10 kg, which is why plexiglass was used. The demonstration was executed to showcase the usefulness of the AR application rather than the robot’s, which is why the material and model are not as important. It is important to clarify that the robot control was adapted to the different model types easily, with the only difference being that the UR10’s URDF and materials are imported to the Unity application instead of the KR60, proving that the application functionalities are essentially model agnostic.

In Figure 17b, the safety zones are shown. The inner red circle represents the maximum range of the robot, which is a forbidding position for the operator to be in when the robot is moving. The outside yellow circle is a warning distance.

FIGURE 17 (a) Setup for AR glasses physical demo with UR, (b) safety zones visualization for AR glasses physical demo with UR, (c) manual guidance visualization for AR glasses physical demo with UR, (d) nut assembly visualization for AR glasses physical demo with UR, simulation of AR instructions for barrier installation on the road: (e) localization for simulation of AR instructions for barrier installation on the road, (f) robot action visualization for simulation of AR instructions for barrier installation on the road, (g) manual guidance instruction for simulation of AR instructions for barrier installation on the road, and (h) screwing barrier instruction for barrier installation on the road.

As mentioned above, several visualizations aid the operator in their work. For example, in Figure 17c,d, the visualization of how the operator must move the robot manually to place the barrier as well as the visualization of the nuts and the tool to screw the barrier on the pillars is shown. These visualizations disappear seamlessly when the operator gets close to them, so they do not obstruct the operator’s view when they need to assemble a part with their hands.

The most important capability that AR provides to the operator is the control of the intervention. Using voice commands, the operator can command the operation to begin and the robot to move; and they can command specific operations, such as the gripper to release, the camera to begin detection of a specific entity, and so forth.

Aside from the demonstrator above, the application was deployed on the road to test localization and get a first view of how the future deployment will look as shown in Figure 17e–h.

From the experiment with the UR10e robot, the simulations, and the research of the current method for barrier installation, compared with the proposed method, we extracted the following deductions. In the current manual method, the barrier installation needs two operators, in order to hold the barriers to the pillars and maybe a third one to screw it on the pillars, although this can be done by the operators who hold it with extra effort. With the proposed method, one operator can guide the barrier, using the robot and AR to control it, removing at least half of the workforce needed on the road, reducing the volume of people in dangerous zones. For the operator who stays on the road, the safety zone visualization helps them stay safe from the robot, while the information from V2X provides an awareness of the road, further enhancing their safety.

To test the hypothesis for safety and efficiency, in the beginning, we used two operators. Both operators were inexperienced with the intervention process. Both attempted to complete the intervention without AR at first, only with preliminary instructions from a trainer on the sequence of the process. They did not manage to complete the intervention at first since they were not familiar with it, and they did not have access to user-friendly robot control interfaces that the AR application provides, having to move the robot using its controller which was time-consuming. Then both inexperienced users were provided with the AR application with instructions and managed to finish the intervention in 2.5 min on average between them. Then they repeated the process, becoming familiar enough to be considered experienced. In this case, they tried again to complete the intervention without the AR application and managed it at approximately the same time as their previous run when they were inexperienced with AR enhancement. The inefficient robot control without the AR interfaces was the factor that delayed them. Then they were given AR again, which provided them with robot-control interfaces, which eliminated the last flaw, allowing them to complete the intervention 1 min faster. These observations combined with knowledge from the research of Section 2 as well as simulations done only virtually with the AR devices led us to the qualitative diagram of Figure 18. As a qualitative diagram, there are no values in the vertical axis since the different heights of the bars show comparable magnitude between the different levels of the experiment.

The results of the experiment were expected, providing further proof of the logical conclusions of the methodology. It is natural that inexperienced operators who have no prior knowledge of a specific process that has several steps and requires the use of specific hardware to not be able to complete it with only rudimentary instructions. AR instructions that explain every step and also provide a user-friendly interface for hardware control that requires no prior knowledge from the user provide both knowledge on how to complete the process and the necessary functionalities to do so therefore is the enabling factor for the completion of the process by inexperienced operators. The advantages that instructions provide are of course less prominent when the operator knows the complete process by memory and, therefore, can be characterized as experienced. This can be considered a weakness of the proposed method, although it is conceptually targeted at inexperienced operators. Also, with the rapid changes in the state of the art, new methods are introduced in industry; therefore, a previously experienced operator can be considered inexperienced with the new methods and may require the proposed method's assistance. An experienced operator can also handle the machinery necessary. The advantage of the proposed method in this case is that the interfaces created by the methodology for the manipulation of the robot are more intuitive and easier to use than the default manipulation method.

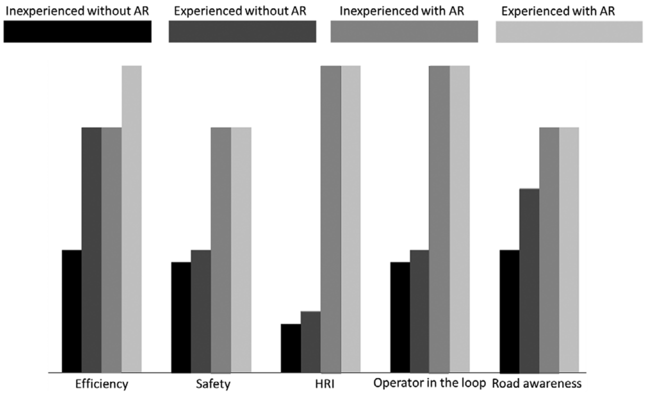


FIGURE 18 The effects of the application on operators.

6 | CONCLUSION AND FUTURE WORK

This paper discusses a methodology for support of road maintenance operators based on AR. The method was designed for specific interventions in road maintenance, yet it is developed in such a way that it is easily adaptable for more use cases. The methodology consists of the following functionalities: (a) HRC and system commands, (b) environment awareness enabled by information feedback and V2X, and (c) visualization of instructions, applied in the dynamic and dangerous environment of the highway. The results of the study support, (1) the hypothesis of the rise in efficiency due to the effect of the HRC, that is enabled by AR, has in the operation, as well as the enhancement of the operator from instructions; (2) the hypothesis of the enhancement of safety since the AR application reduces the necessary workforce, meaning less people in dangerous environments, reduces the cycle time since the operators are more efficient, and provides safety visualizations.

While the proposed method is adjusted to road maintenance operations, it is developed to be able to be adjusted on any operation that takes place in a dynamic environment and includes an operator and a robot that the operator must collaborate with to perform their operations. The technologies that are specific to the road industry may limit the application itself to road maintenance, but the principles of them do not. For example, V2X information is road information, but as it regards the AR part, what it provides to the operator is a better awareness of their environment, and this, as a concept, is necessary for any operator who works in dangerous, dynamic environments, no matter the source of the information. All the functionalities that are described in the paper can be attributed to principles, which when combined create an AR-based operator support method that can be adapted to a plethora of industrial or academical operations. The road maintenance sector works as the proving ground for this method.

The limitation of this study is the lack of evidence in real applications on the highway. Future work entails the integration of the framework with a real mobile robot platform and the validation of its effect by actual robot operators, which will be reported in later manuscripts. In this validation, any complexities of highway maintenance that have not been taken into consideration by the current approach will also be addressed.

ACKNOWLEDGMENTS

This work has been partially funded by the European Commission Research project "OMICRON—Towards a more automated and optimized maintenance, renewal and upgrade of roads by means of robotized technologies and intelligent decision support tools" (Grant Agreement: 955269) (https://omicronproject.eu//).

REFERENCES

Baroroh, D. K., Chu, C. H., & Wang, L. (2021). Systematic literature review on augmented reality in smart manufacturing: Collaboration between human and computational intelligence. *Journal of Manufacturing Systems*, *61*, 696–711.

Chitta, S. (2016). MoveIt!: An introduction. In A. Koubâa (Ed.), *Robot operating system (ROS): The complete reference* (pp. 3–27). Springer.

Chryssolouris, G. (2006). *Manufacturing systems: Theory and practice* (2nd ed.). Springer.

Crick, C., Jay, G., Osentoski, S., Pitzer, B., & Jenkins, O. C. (2017). Rosbridge: ROS for non-ROS users. In H. I. Christensen & O. Khatib (Eds.), *Robotics research: The 15th international symposium ISRR* (pp. 493–504). Springer International Publishing.

Dimitropoulos, N., Togias, T., Zacharaki, N., Michalos, G., & Makris, S. (2021). Seamless human–robot collaborative assembly using artificial intelligence and wearable devices. *Applied Sciences*, *11*(12), 5699.

Dimitropoulos, N., Togias, T., Michalos, G., & Makris, S. (2021). Operator support in human–robot collaborative environments using AI enhanced wearable devices. *Procedia CIRP*, *97*, 464–469.

Eiriksdottir, E., & Catrambone, R. (2011). Procedural instructions, principles, and examples: How to structure instructions for procedural tasks to enhance performance, learning, and transfer. *Human Factors*, *53*(6), 749–770.

Eskandari Torbaghan, M., Kaddouh, B., Abdellatif, M., Metje, N., Liu, J., Jackson, R., Rogers, C. D. F., Chapman, D. N., Fuentes, R., Miodownik, M., Richardson, R., Purnell, P., & Purnell, P. (2020). Robotic and autonomous systems for road asset management: A position paper. *Proceedings of the Institution of Civil Engineers-Smart Infrastructure and Construction*, *172*(2), 83–93.

Ghosh-Dastidar, S., & Adeli, H. (2006). Neural network-wavelet microsimulation model for delay and queue length estimation at freeway work zones. *Journal of Transportation Engineering, ASCE*, *132*(4), 331–341.

Hietanen, A., Pieters, R., Lanz, M., Latokartano, J., & Kämäräinen, J. K. (2020). AR-based interaction for human-robot collaborative manufacturing. *Robotics and Computer-Integrated Manufacturing*, *63*, 101891.

Jiang, X., & Adeli, H. (2004). Object-oriented model for freeway work zone capacity and queue delay estimation. *Computer-Aided Civil and Infrastructure Engineering*, *19*(2), 144–156.

Karelina, M. Y., Vasiliev, A. V., Guly, V. V., Podgorny, A. V., & Erpulev, V. A. (2022). Robotic systems in road construction. *2022 Systems of Signals Generating and Processing in the Field of on Board Communications*, Moscow, Russian Federation (pp. 1–4).

Karim, A., & Adeli, H. (2003). CBR model for freeway work zone traffic management. *Journal of Transportation Engineering, ASCE*, *129*(2), 134–145.

Katsamenis, I., Bimpas, M., Protopapadakis, E., Zafeiropoulos, C., Kalogeras, D., Doulamis, A., Doulamis, N., Montoliu, C. M.-P., Handanos, Y., Schmidt, F., Ott, L., Cantero, M., & Lopez, R. (2022). Robotic maintenance of road infrastructures: The HERON project. *Proceedings of the 15th International Conference on Pervasive Technologies Related to Assistive Environments*, Aalborg, Denmark (pp. 628–635).

Kyjanek, O., Al Bahar, B., Vasey, L., Wannemacher, B., & Menges, A. (2019). Implementation of an augmented reality AR workflow for human robot collaboration in timber prefabrication. Proceedings of the 36th International Symposium on Automation and Robotics in Construction, ISARC, Banff, Canada (pp. 1223–1230).

Lambrecht, J., Kästner, L., Guhl, J., & Krüger, J. (2021). Towards commissioning, resilience and added value of Augmented Reality in robotics: Overcoming technical obstacles to industrial applicability. *Robotics and Computer-Integrated Manufacturing*, *71*, 102178.

Lee, A., Lee, J. H., & Kim, J. (2016). Data-driven kinematic control for robotic spatial augmented reality system with loose kinematic specifications. *ETRI J*, *38*, 337–346.

Lindblom, A. (2020). *A study of networking performance in a multi-user VR environment: Using Unity and the Mirror library* [Bachelor thesis, Luleå University of Technology].

Liu, H., Zhang, Y., Si, W., Xie, X., Zhu, Y., & Zhu, S. C. (2018). Interactive robot knowledge patching using augmented reality. 2018 IEEE

International Conference on Robotics and Automation (ICRA), Brisbane, Australia (pp. 1947–1954).

Lotsaris, K., Fousekis, N., Koukas, S., Aivaliotis, S., Kousi, N., Michalos, G., & Makris, S. (2021). Augmented reality (AR) based framework for supporting human workers in flexible manufacturing. *Procedia CIRP*, *96*, 301–306.

Lotsaris, K., Gkournelos, C., Fousekis, N., Kousi, N., & Makris, S. (2021). AR based robot programming using teaching by demonstration techniques. *Procedia CIRP*, *97*, 459–463.

MacHardy, Z., Khan, A., Obana, K., & Iwashina, S. (2018). V2X access technologies: Regulation, research, and remaining challenges. *IEEE Communications Surveys & Tutorials*, *20*(3), 1858–1877.

Makris, S. (2020). *Cooperating robots for flexible manufacturing*. Springer. https://doi.org/10.1007/978-3-030-51591-1

Makris, S., Pintzos, G., Rentzos, L., & Chryssolouris, G. (2013). Assembly support using AR technology based on automatic sequence generation. *CIRP Annals*, *62*(1), 9–12.

Malek, K., Mohammadkhorasani, A., & Moreu, F. (2023). Methodology to integrate augmented reality and pattern recognition for crack detection. *Computer-Aided Civil and Infrastructure Engineering*, *38*(8), 1000–1019.

Maurtua, I., Ibarguren, A., Kildal, J., Susperregi, L., & Sierra, B. (2017). Human–robot collaboration in industrial applications: Safety, interaction and trust. *International Journal of Advanced Robotic Systems*, *14*(4), 1729881417716010.

Michalos, G., Karagiannis, P., Makris, S., Tokçalar, Ö., & Chryssolouris, G. (2016). Augmented reality (AR) applications for supporting human-robot interactive cooperation. *Procedia CIRP*, *41*, 370–375.

Ni, D., Yew, A. W. W., Ong, S. K., & Nee, A. Y. C. (2017). Haptic and visual augmented reality interface for programming welding robots. *Advances in Manufacturing*, *5*, 191–198.

Pai, Y. S., Yap, H. J., Dawal, S. Z. M., Ramesh, S., & Phoon, S. Y. (2016). Virtual planning, control, and machining for a modular-based automated factory operation in an augmented reality environment. *Scientific Reports*, *6*, 27380.

Ong, S. K., Yew, A. W. W., Thanigaivel, N. K., & Nee, A. Y. (2020). Augmented reality-assisted robot programming system for industrial applications. *Robotics and Computer-Integrated Manufacturing*, *61*, 101820.

Quigley, M., Conley, K., Gerkey, B., Faust, J., Foote, T., Leibs, J., Berger, E., Wheeler, R., & Ng, A. Y. (2009). ROS: An open-source Robot Operating System. *ICRA Workshop on Open Source Software*, Kobe, Japan (Vol. *3*, No. 3.2, p. 5).

Ren, B., Luo, X., Li, H., Chen, J., & Wang, Y. (2022). Gait trajectory based physical human-robot interactive controller for construction workers' lower limb exoskeletons. *Computer-Aided Civil and Infrastructure Engineering*, *37*(5), 558–572.

Rentzos, L., Papanastasiou, S., Papakostas, N., & Chryssolouris, G. (2013). Augmented reality for human-based assembly: Using product and process semantics. *IFAC Proceedings Volumes*, *46*(15), 98–101.

Sadhu, A., Peplinski, J. E., Mohammadkhorasani, A., & Moreu, F. (2023). A review of data management and visualization techniques for structural health monitoring using BIM and virtual or augmented reality. *Journal of Structural Engineering*, *149*(1), 03122006.

Sheta, A., & Mokhtar, S. A. (2022). Autonomous robot system for pavement crack inspection-based CNN model. *Journal of Theoretical and Applied Information Technology*, *100*(16).

Si-Mohammed, H., Petit, J., Jeunet, C., Argelaguet, F., Spindler, F., Evain, A., Roussel, N., Casiez, G., & Anatole, L. (2018). Towards BCI-based interfaces for Augmented Reality: Feasibility, design and evaluation. *IEEE Transactions on Visualization and Computer Graphics*, *26*, 1608–1621.

Shim, S., Lee, S.-W., Cho, G.-C., Kim, J., & Kang, S.-M. (2023). Remote robotic system for 3D measurement of concrete damage in tunnel with ground vehicle and manipulator. *Computer-Aided Civil and Infrastructure Engineering*, *38*(15), 2180–2201.

Siatras, V., Athanasopoulou, L., Alexopoulos, K., Stavropoulos, P., & Mourtzis, D. (2021). Applying the teaching factory paradigm and Augmented Reality technology for operator training in assembly operations. *Proceedings of the Conference on Learning Factories (CLF)*, Singapore.

Tarek, H., & Marzouk, M. (2022). Integrated augmented reality and cloud computing approach for infrastructure utilities maintenance. *Journal of Pipeline Systems Engineering and Practice*, *13*(1), 04021064.

Wang, X., Ong, S. K., & Nee, A. Y. C. (2016). Multi-modal augmented-reality assembly guidance based on bare-hand interface. *Advanced Engineering Informatics*, *30*(3), 406–421.

Zhu, D., & Veloso, M. (2016). Virtually adapted reality and algorithm visualization for autonomous robots. In *RoboCup 2016: Robot World Cup XX 20* (pp. 452–464). Springer International Publishing.

How to cite this article: Bavelos, A. C., Anastasiou, E., Dimitropoulos, N., Oikonomou, G., & Makris, S. (2024). Augmented reality-based method for road maintenance operators in human–robot collaborative interventions. *Computer-Aided Civil and Infrastructure Engineering*, *39*, 1077–1095. https://doi.org/10.1111/mice.13185