Received: 15 April 2024 | Accepted: 27 August 2024

DOI: 10.1111/mice.13337

INDUSTRIAL APPLICATION

COMPUTER-AIDED CIVIL AND INFRASTRUCTURE ENGINEERING WILEY

Virtual reality-based dynamic scene recreation and robot teleoperation for hazardous environments

Angelos Christos Bavelos | **Efthymios Anastasiou** | **Nikos Dimitropoulos** | **George Michalos** | **Sotiris Makris**

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

Correspondence
Sotiris 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
OMICRON under H2020 SOCIETAL CHALLENGES - Smart, Green And Integrated Transport, Grant/Award Number: 955269

Abstract

Virtual reality (VR) technology is increasingly vital in various sectors, particularly for simulating real environments in training and teleoperation. However, it has primarily focused on static, controlled settings like indoor industrial shopfloors. This paper proposes a novel method for remotely controlling robots in hazardous environments safely, without compromising efficiency. Operators can execute tasks from remote locations ensuring continuity regardless of distance. Real-time efficiency is achieved by updating the virtual environment from on-site sensors and mirroring the real environment, utilizing 3D reconstruction, Google Images, and video streams. Communication between VR and the remote robot is facilitated through a remote robot operating system connection. The efficacy of this concept will be validated through real road maintenance interventions.

1 | INTRODUCTION

Virtual reality (VR) has been proven in research as a tool with great potential for enhancing manufacturing, with benefits such as optimization of resource utilization (Chryssolouris, 2006) and cooperation with robots for larger product variability (Makris, 2020). These advantages can be transferred to other industries as well. VR is a simulated experience that uses 3D near-eye displays and pose tracking to immerse the user in a virtual environment (VE). The interaction of the user and the VE affects the VE, but these interactions can be bridged with other digital systems, such as robotics, to mimic the interactions in reality. VR-based remote robot control provides an intuitive way of human–robot collaboration for users with no experience in robotics, applicable for any industry that applies a hybrid model consisting of both robots and humans and requires interaction.

A great advantage that VR in combination with robotics can offer to workers is the enablement of remote operations, which is most useful for operators who work in hazardous environments, due to environmental or machinery related dangers. While these operations have specific characteristics that make them hazardous, teleoperation does not need to account for each specific hazard since it tackles the issue at the root: human presence in hazardous environments. Of course, this requires the operation to be able to be executed by robots and to not absolutely require the direct handling of a human, only their cognitive abilities. With the constant advances in technology, those operations become more and more common. An example of such operations is road maintenance. In high-traffic highways, the maintenance operators are in danger even when they work in closed-off lanes, as the danger that a car may stray into the closed-off lane is ever present. Statistics show that the road worker death toll in the EU in 2020 was 3355 people, showing increase from the year before by 53 people. Road maintenance is primarily executed manually. Remote operations until now have been focused mainly on monitoring systems for the road condition (Miyamoto, 2021).

The obvious solution to the problem of hazardous environments for workers and the way to reduce the number of accidents is to remove the road workers from the hazardous environment. While robots can take the place of the human operator in some cases, even working better than the human due to their status as a tireless machine, in specific operations that require decision making and experience, the human mind cannot often be replaced. Therefore, the ideal situation is to have the operator away from hazards but in control of the operation executed by robots.

Robotics in road operations is a recent topic. In Katsamenis et al. (2022) the authors proposed a robotic vehicle supported by autonomous drones to coordinate road maintenance works. The authors in Karelina et al. (2022) propose an autonomous system for robotic control of the working body of a bulldozer when erecting a road on the canvas, while Eskandari et al. (2020) propose an automated robotic system for road crack-sealing by using 3D printing techniques.

This paper proposes a method for remote teleoperation of robots in hazardous environments regardless of distance, effectively removing the operator from them and negating the chance of accidents. The objectives are to establish full operator awareness of the remote environment, as well as achieve teleoperation independent from distance. The proposed method is validated in a road maintenance case study, which is developed to be applied in dangerous high traffic highways.

The manuscript is organized as follows: Section 2 comprises the literature review, while Section 3 provides a description of the proposed methodology. Section 4 reviews the implementation of the discussed method. In Section 5, a use case for the validation of the methodology and the results are presented, while in Section 6, the results are discussed. The last section is dedicated to summarizing the conclusions from the study as well as suggesting the authors' future work.

2 | LITERATURE REVIEW

2.1 | VEs for simulations

Extended reality (XR) in general is most frequently used in research for training of operators, in combination with other training concepts such as the Teaching Factory Paradigm (Siatras et al., 2021). VR can be used for training new operators with great effect as seen in Dimitropoulos et al. (2020) where the authors introduced a method for training operators in a VE replicated after the real cell that the operators will work, with the proper interactions between the operator and the objects they must use to perform the process and between the objects themselves. Also, in Barkokebas et al. (2019), the data from training regarding completion time and ergonomics are collected and then used as input for production managers to improve the process via changes in the process itself or the layout. In Lanz (2022), the authors analyze the technical feasibility and industrial readiness of VR for safety training in the manufacturing sector, with the results indicating that VzR is valued over the traditional methods. In Cutini et al. (2023), the authors utilize a VE for agriculture training, developing interactions for tractors, farm equipment, and human–machine interfaces, describing how the design choices enabled the creation of a precision agriculture simulator. This showcases the great potential that VR possesses as it regards faithful adaptation of real interactions digitally. An interesting use of VR is reported in Kim et al. (2023) where it was used to simulate the road environment for the operators and judge the possibility of road accidents happening due to them becoming habituated to the warning alarms of vehicles. VR is a great tool for offline training, owing to its ability to replicate the operational environments and functions very faithfully, but the limitation of these methods is that after the operator has moved on from training to application, and they are no longer of use in dynamic environments due to their rigidity. This method that is proposed, intended for the on-line part of the operation, combines static visualizations, like the ones used for training, with dynamic elements. VR can also be used to create a digital twin for process monitoring and control, as in Z. Zhang et al. (2022), where Artificial Intelligence (AI) is also used to analyze the sensor data that comprise the digital twin. VR is even used in civil infrastructure, for example, in Luleci et al. (2022) where a VR application is developed to bring the structure and structural health monitoring data from the field to the office and enable the inspection from several experts who will be able to visit bridge structures virtually. Li et al. (2021) discuss the virtual trial assembly, which is a process that simulates the physical trial assembly of a structure in a VE and is very useful at reducing the cost of assembly of complex structures. J. Zhang et al. (2014) propose a graphic information model to enhance virtual construction applications. These simulations can identify problems that could happen in the actual construction and thus avoid them.

2.2 | VR for human–robot interaction

Robotics have long been used for operations that are too dangerous for humans. Mobile robots are especially efficient, being able to move freely throughout the entirety of the operation's shopfloor, their flexibility often making the direct presence of the human operator unnecessary. This concept has been gaining focus in recent research. While automated processes are usually more efficient than manual processes performed by the human, especially when involving hazardous operations and heavy machinery, it is the collaboration between human and machines such as 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 together with the robot's advantages, and it is a key factor for the development of factories of the future (Maurtua et al., 2017). While in these cases, teleoperation is superior to full automation, unstructured and complex environments increase the collision risk, which means that the teleoperation system should include collision risk perception and security strategies to improve safety and efficiency as reported in Peng et al. (2023). VR is a great enabler of human–robot collaboration. Togias et al. (2021) explain how VR can be used to reprogram industrial robots to achieve flexibility in production without the operator's physical presence in the shopfloor. The authors used robot operating system (ROS) to connect to the real robot from inside a VE interface and move it through virtual end effector manipulation. The limitation of their method is that it works only with local networks, providing very limited teleoperation capabilities, in which the operator must be close to the robot, with the option of being in a different room, as long as they are in the same network. Additionally, there is no environment reconstruction, or real-time environment monitoring, working only for static environments with no dynamic elements in them, although they intend to fix that in future works. In Zhou et al. (2020), a system is proposed that reconstructs the environment of the robot in VR through deep learning scene reconstruction and performs teleoperation of the robot according to the VE, although they also do not suggest a truly remote connection, working with the local connection of ROS. 3D reconstruction for teleoperation has also gained the attention of specialists for mining operations, based on the reconstruction of underground mining environments with the use of point clouds. This is reported in Kamran-Pishhesari et al. (2024) as a way to avoid the hazardous environment of underground mining. In Perez et al. (2019), the working area of the robot is reconstructed in VR from 3D point clouds, and the immersed operator can simulate different positions of the robot, calculate the singularities, verify the reachability, and study any possible collisions, although the teleoperation is limited, in the sense that the operator must be in the same workplace as the robot, and although obstacle avoidance is considered, the method proposed does not ensure it, leaving it to the operator's care according to visualizations. Teleoperation with sensor feedback is also reported in Solanes et al. (2022), for a non-industrial mobile TurtleBot, as it regards spatial navigation.

2.3 | Other methods for robot teleoperation

VR is not the only XR technology that is used for robot manipulation, as augmented reality (AR) is also well known in this capacity and is a well-researched topic. The authors in Arevalo et al. (2021) propose a method that provides visual cues about the position of the gripper to the operator via AR in order to improve distance perception, while in Pan et al. (2021), the authors created an AR-based teleoperation teaching system using red green blue-depth (RGB-D) imaging and attitude teaching, allowing even non-experienced operators as it regards to robot programming, to use their system easily. The limitation of AR-based methods is, again, that in this case, the operator must be in the operational environment and cannot be completely out of danger. Other technologies have also been researched for teleoperation, such as in Zhu et al. (2022), where a haptic suit was used to generate haptic feedback for the operator that corresponds to the bottom and up sides of a snake robot, combined with VR visualizations, for the effective manipulation of snake robots. This method was developed to avoid the simple camera feedback for teleoperation that fails to present the wanted view from all angles and thus misses a lot of information. Fishel et al. (2020) proposes tactile telerobots for dangerous and inaccessible tasks, utilizing tactiles to teach the robot fingers to work with near the same performance as their bare hands, although the method lacked a perception system to allow for true teleoperation, regarding it as future work. Other methods for teleoperation include brain–machine interfaces (BMIs), such as the one presented by Zhao et al. (2023), in which a BMI is utilized to induce an electroencephalogram to control the motion of a wheeled robot. This approach requires visual feedback to offer the operator awareness of the environment, through a camera, in a non-unified interaction interface, in contrast with a VR-based approach, where several perception techniques can be combined and integrated to one representation.

2.4 | Advantages of VR for teleoperation and gap to be bridged

A great advantage of VR is that it is a rich sensory experience, with the greatest potential for interactivity, keeping the user engaged and stimulated, compared to more mundane and one-dimensional methods. An important

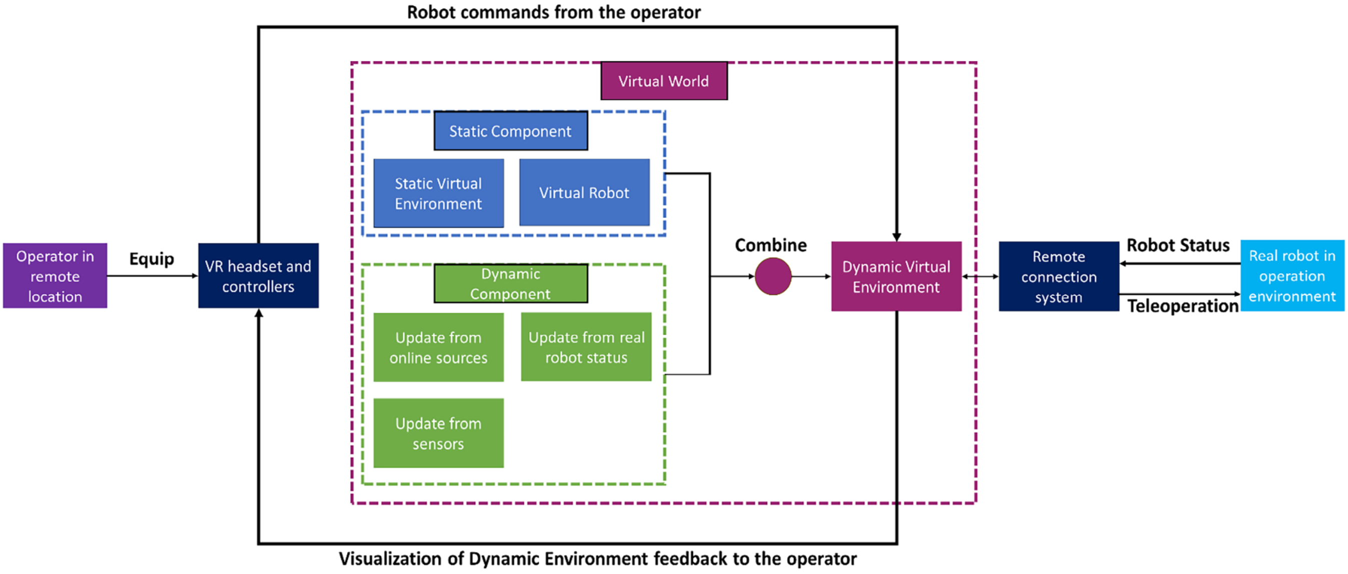

FIGURE 1 Virtual reality (VR) teleoperation approach.

concept for human–robot collaboration interfaces, which is appliable to VR interfaces for teleoperation is that they should be user-friendly. To be user-friendly, an interface must ascertain the user's needs and ensure reliability to appease the user's mistrust as reported in Shneiderman and Plaisant (2010).

All the mentioned approaches showcase the superiority of XR methods and specifically VR, as a visualization media enabling teleoperation in comparison to simple camera view feedback, which misses a lot of information. But even the aforementioned methods miss an opportunity to take VR, and in general XR teleoperation to even more complex operations, such as operations in fully dynamic in nature environments. For most of them, teleoperation is quite limited in range, requiring the operator to be in the same building as the robot, which in many cases is not possible. Such is the case with road maintenance interventions for highways, where the operator would be in danger in any place on the highway, demanding unlimited distance of teleoperation. Road maintenance is a dangerous job for the operator, particularly in high traffic areas, in addition to being physically demanding and raising ergonomic concerns. Safety is a big concern for road operators due to the high speed of the vehicles on highways, making any possible contact usually fatal.

3 | METHODOLOGY

3.1 | Dynamic VE

In order to enable safety and efficiency in hazardous operations, a methodology for teleoperation based on VR technologies is proposed. According to the proposed method, VR will be used when the operator is in a remote location, as it will work independently of the distance between the VR system and the robot. The remote operator will be aware of the environment of the robot through data input from on-board sensors. This concept is shown in Figure 1.

For the safe teleoperation of a robot, the operator must be aware of the surrounding environment and possess the necessary information about the parameters of the operation. This is achieved with input data from the real, dynamic environment of the robot. The dynamic environment is created from a static VE that is enhanced by dynamic data exchange. The static environment comprises virtual objects that represent assets that are certain to be in the environment by the nature of the operation. For example, if the operation requires a machine by default, the machine is part of the static environment. The robot is also a default asset in any robotic operation. Knowledge of the static environment is not enough for a safe operation when the environment changes in operational real-time. To add a dynamic component to the environment, data input is needed. These data are received from sensors on board the real robotic platform and also from online information from the web, such as Google Images. The sensors that are used can be visual sensors, such as depth cameras, but part of the information can also be received by other sensors, such as laser scanners, as shown in Dias et al. (2006), where real scenes are reconstructed using a modified 2D laser range finder to work as a 3D laser scanner. The sensors need to be able to output 3D information. The data are then processed, to provide useful information to the operator. 3D information, such as a mesh, is

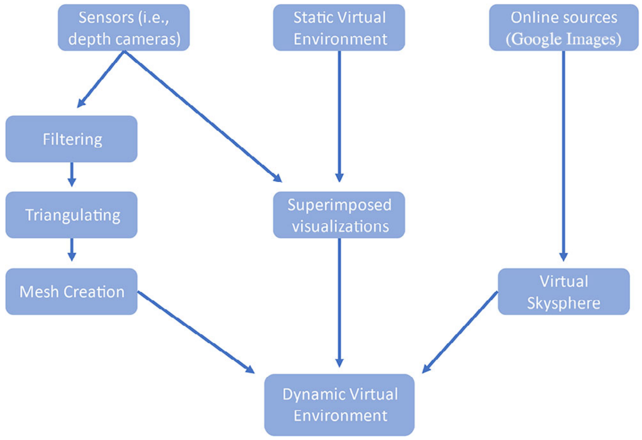

FIGURE 2 Dynamic environment update.

most useful for the operator to be able to see the objects in the environment. In order to create a mesh, first of all, the 3D point cloud is received from the depth camera. Then, filters are used to reduce the information, especially the noise. First, a passthrough filter cuts off the information beyond the range that interests the operator, specifically in the axis that represents the depth of the camera. Due to the range of the robot, the distance that was chosen was 2 m, although this is dependent on each use case's area of interest. Voxel grid downsampling is applied to reduce the point cloud density, adjusting the leaf size according to the point cloud (Miknis et al., 2015). Then a statistical outlier removal filter (Balta et al., 2018), which removes the points that are furthest from their neighbors, is used for the removal of noisy points. The number of neighbors to analyze for each point is set to 50 and the standard deviation multiplier to 1, meaning that all points that have a distance larger than 1 standard deviation of the mean distance to the query point will be marked as outliers and removed. A radius outlier removal filter was also considered, which removes points that have less neighbors than a set number in a specific radius, but the statistical one provided better results. After the processing, a Greedy Projection Triangulation takes place, which outputs a set of vertices and triangles that are transferred as inputs to the visualization software that creates the mesh. This way, before or during the intervention, the remote operator has a dependable view of the surrounding environment of the robot. The mesh is also utilized as an input by the motion planners for the robot's obstacle avoidance as it will be explained in the next chapter.

Additionally, even 2D images can show specific views of the environment with important details for the operation, as they have traditionally been used in teleoperation, they can be added to a more complex method. All this information can be superimposed in a static VE of the operation's space, in a combination that creates a dynamic model of the real space in the VE, essentially a digital twin that contains all the necessary data for the operator to make decisions remotely. The superimposition of the 2D camera view can be done according to the comfort of the operator. The image stream can either appear to match the view of the real camera as it regards position, or it can appear in any position that the operator desires. The base of the VE can be either a pre-modeled default environment that contains a skeleton of what is certain to exist in the operation's environment or, in the case of outdoor operations, an environment extracted from Google Images, from the exact point of the operation, so the combination with the dynamic parts will create an awareness for the operation location that is complete. In order to create an environment from Google Images, a 360° skysphere is shaped from the geographical location that the user can provide. In this way, any outdoor environment can be modeled with a high quality of visualization. The process is summarized in Figure 2.

3.2 | Robot manipulation

As it regards the manipulation of the real robot, the real robot is linked to the virtual one in the VE. Therefore, the virtual robot's joint positions are updated as the real robot's joint positions change in real time. Now that the operator can monitor the status of the robot in VR, they can issue remote commands, which are executed by the real robot and also visualized in VR in real-time, seeming as if the operator is commanding the virtual robot. It would be more accurate to say that the virtual robot is merely visualizing the changes in the real one. Of course, before any execution takes place, the operator can request a visualization—in VR—of the trajectory before it happens and either approve it or request a new one. This is an extra safety step since the operator is able to see the movement of the robot before it happens and if it is unsafe for any reason, cancel it and request a better one. After the approval, the real robot executes the movement and the virtual robot is updated in real time.

In every operation, there are several ways to move a robotic arm, according to the parameters of the operation, although in complex operations, several types of movement may be necessary. In the proposed method, the robot can move in several ways: First, by changing its joint values from VR, changing each joint separately or sending a command for all joints. The robot can also move by setting a desired position of the end effector and solving the inverse kinematics to reach the desired position. Finally, the operator is able to move the end effector in a linear way, in the same way that they can move it from a teach pendant. These types of movements range in precision and speed,

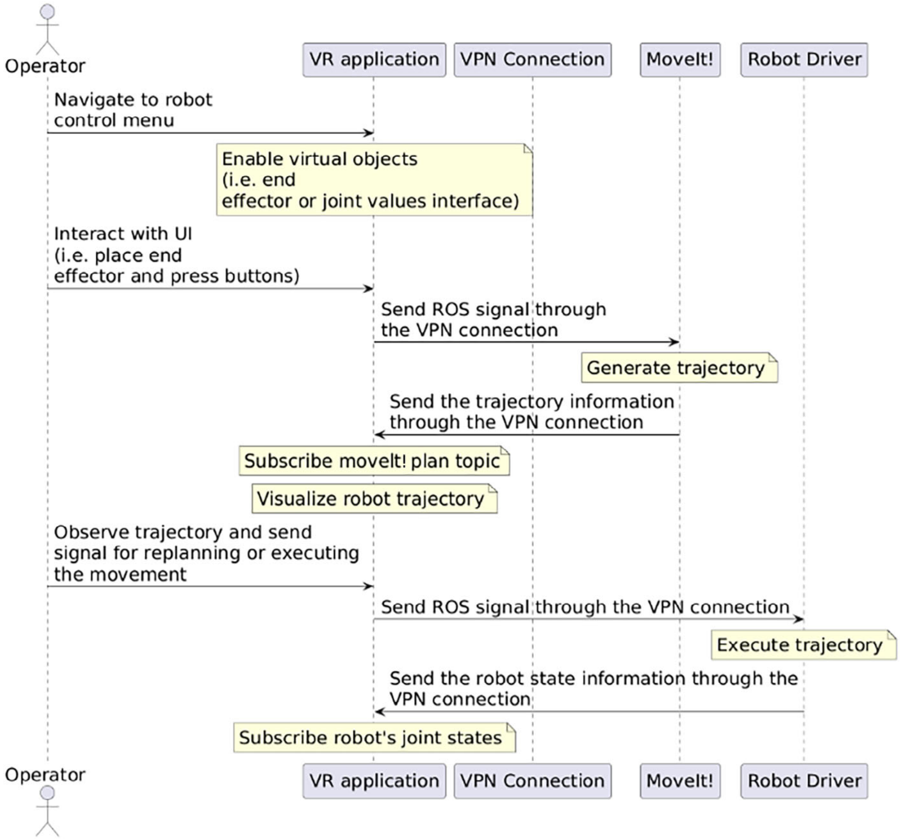

FIGURE 3 Robot control sequence diagram.

and the operator can choose according to the parameters of the operation.

The remote communication of robot and VR and also of remote sensors and VR is achieved based on the Internet, meaning it can work effectively anywhere where there is an Internet connection. This allows for unlimited distance since it does not matter when in relation to the real robot the operator will be, as long as they have access to the Internet. There is a choice of virtual private network (VPN) or port-forwarding, each with its own merits. The software and methods of the remote connection are analytically explained in the next section. As it regards the communication between the operator and the VE, it is done via the interfaces of the VR headset. For communication with the robot, the ROS was used, an open-source middleware that comprises a set of software libraries that is primarily used to build robotic applications. This will be further expanded in the next section. Figure 3 presents a sequence diagram of the robot control functionality.

4 | IMPLEMENTATION

This section aims to explain the implementation of the approach covered in the previous section. For the development of the VR functionalities, 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 application programming interface (API) based on C#, with which scripts with various functions can be created and attached to the objects to define their behavior. This implementation was developed using version 2019.4.17 of Unity3D. The headset that was used for the development was an HTC Vive Pro. For communication with the robot, as mentioned in the previous section, the ROS was used, more specifically the Noetic distribution. Below, the specific development details for each module are analyzed.

4.1 | Dynamic environment update

4.1.1 | Mesh creation

The dynamic VE is created by adding sensor data to the static scene, from on-board sensors in the real environment. The most useful of those sensors for this specific function are sensors that output 3D information. By utilizing this information, the real environment can be

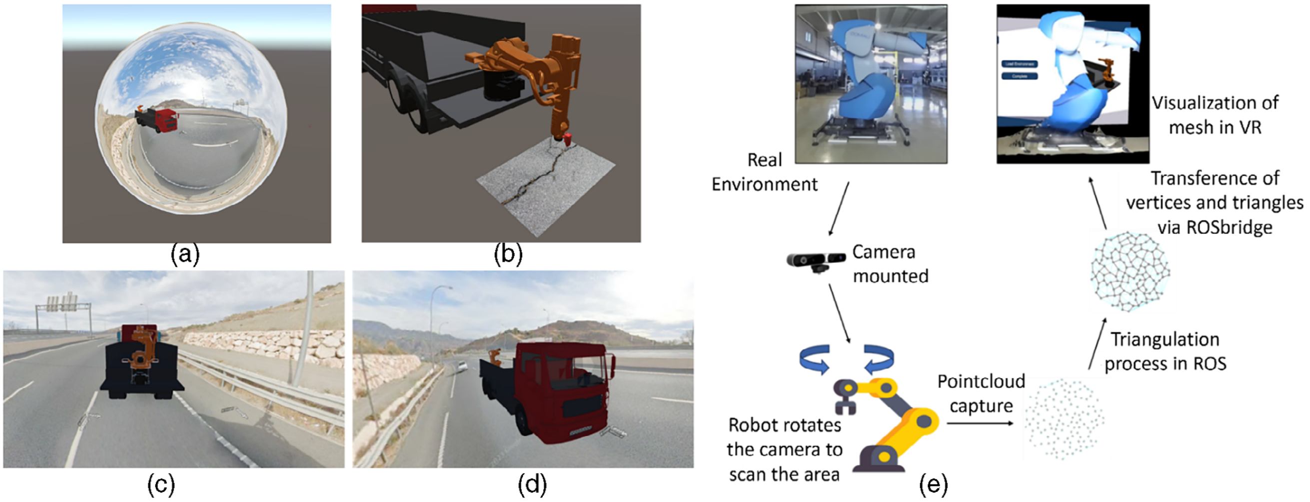

FIGURE 4 VR dynamic environment: (a) Virtual skysphere, (b) image visualization, (c) Google Street View images rear view, (d) Google Street View images front view, and (e) environment reconstruction process.

reconstructed in the VE. Those sensors can be camera or laser sensors or any sensor that can extract point clouds from their view. The concept of the reconstruction is shown in Figure 4e.

The camera captures a point cloud of the area and passes the point cloud to ROS. ROS has drivers for various sensors and can thus communicate with the devices to publish "topics" that contain the information in the form of messages. The message for the point cloud in ROS ("sensor_msgs/PointCloud2.msg") contains several fields including a header declaring time of sensor data acquisition and the coordinate frame ID, but the actual data are transmitted as an uint8 array that contains all the points of the cloud. Then, a developer's node "subscribes" to the point cloud topic, receives the message, extracts the points, and then begins the processing. In this case, a script receives the point cloud data and performs a triangulation of the point cloud to create a mesh as explained in the previous section. But the data received from the camera can be too much, for example, in the experiments done in this paper, about 300,000 points in 3D space. This point cloud is reduced to about 10% of the initial points using filters as mentioned in the previous section, to remove outliers and keep the most "solid" point cloud groups, which constitute objects in the immediate area. The point cloud reduction also aids in the speed of the algorithm since the triangulation of hundreds of thousands of points would be too heavy computationally and would likely not result in the desired outcome, as the noise would be triangulated as well. The triangulation is done in ROS with the help of the point cloud library (PCL). After receiving the point cloud, the process estimates surface normals for each point in the cloud, a crucial step for understanding the geometry of the data. The program integrates the point coordinates and the calculated normals into a new point cloud. To enhance the efficiency of point searches, a KdTree data structure is established. The core of the method involves the application of the Greedy Projection Triangulation algorithm, which utilizes the integrated point cloud and the KdTree for efficient triangulation. The triangulation algorithm computes a polygon mesh representation of the 3D surface and outputs it as a mesh object, comprising vertices, containing the vertices of the mesh, and a triangles list. Once the point cloud is triangulated, the vertices and the relevant triangles are transported to Unity3D where an object is created, with a mesh property that comprises the vertices and triangles received. The vertices that are transferred via the PointCloud2 are in an array that contains for each point the x, y, and z values of the position of the point in the 3D space, as well as intensity and rgb values. The rgb values are transferred to Unity3D and saved as a Texture object, which is essentially a bitmap image that contains information in three dimensions, and is then added as input to the created mesh object. The communication between ROS and software, which is not ROS-based, is performed with ROSbridge, which is used by Unity3D in this case. The 3D reconstruction process is shown in Figure 4e, where a robot and its base are reconstructed as a demonstration.

Before the creation of the mesh, the immediate environment of the robot is unknown; therefore, for the movement of the robot's end effector, which moves the camera to capture the scene, to be without collision, some safety rules are followed, Namely, the movement is very small and can even be only a rotation of the final joint of the robot, which is highly unlikely to produce a collision. Additionally, the camera can already provide a video stream to ensure that the operator has some perspective of the environment already. Finally, the most possible collision from this movement would be with the static environment very close to the robot, for example, the robot's base, which has already been included in the planning scene of the robot.

4.1.2 | Image updates

In the proposed method, an innovative solution for creating VEs that simulate outdoor real environments is proposed. When the location of the environment in which the robot will work is determined, the operator can input the coordinates into the VR application using the developed interfaces. The VR application can then collect street view images directly from the Google Images API at the specified location. Upon inserting the coordinates of the activity area into the VR application, the application can request the corresponding images of the designated location. For each location, the application collects six different images. These images are carefully designed to seamlessly blend together at the edges of a virtual cube, creating a continuous and immersive background. With this technique, the VR application creates a skysphere, which refers to a virtual 3D environment, providing the user with awareness of the real environment at the activity area location.

The robot and any other asset necessary for the faithful representation of the environment are added in the skysphere, and the operator can work inside, as shown in Figure 4a,c,d, with the example of an outdoors highway environment.

Of course, Google Images are not always up to date as they are not updated every day and may be months behind, so there is a chance that the environment would not be in its exact current state, but the skysphere serves to give a general awareness of how the remote location is to the operator and offer familiarity, reducing their uncertainty, giving them a first idea of any topological challenges there may be, and boosting their confidence in the remote operation. The skysphere may also aid in the communication with any on-site workforce, such as the driver of a road maintenance truck. In the future, where technological advances may allow improved update of Google Images, the VE created could be completely accurate. For now, combination with other forms of update such as the mesh creation complete the "image" of the operational environment for the operator.

Aside from Google Images, the operator also has access to image streams from the on-board sensors that will show the environment from specific views, according to the needs of the operation. Usually, these types of sensors refer to depth cameras or laser scanners. The sensors, which have ROS drivers, can publish their camera views to topics. Through a ROSbridge server, the application in Unity can receive the information of these sensors and use it to visualize the real environment on the application's viewport. To achieve this visualization, initially the VR application transforms the information from the sensors to image. After this transformation, the application utilizes the depth information of the sensor to calculate the position of the visualized objects in relation to the sensor. Knowing the exact positions of the sensors within the activity area, the application has the required information to place the images at the correct positions within the VE, enhancing the awareness of the operator. For example, if the real camera is set to look at the floor, then the image view in the VR application will be positioned on the virtual floor, providing a more realistic view of the environment. This example is shown in Figure 4b, where an image of a road crack is projected from the camera mounted on the robot. In this implementation, the camera is placed at a fixed position in relation to the robot. With this way, the application positions the image of the road at the distance calculated by the depth of the camera.

4.2 | Robot control

4.2.1 | Loading the robot in VR

In order to load the virtual robot in VR, the Unified Robot Description Format—an xml representing the robot model's joints and links—of the robot is imported together with the 3D model of the robot. Then this robot can be loaded in the VE scene and can be modified in the link level, by adding scripts that define if each joint is revolute or not, its physics, and scripts that read and write joint states, for the joint states to be updated from ROS and to update ROS from VR. The general concept can be seen in Figure 6a.

4.2.2 | Creating the dynamic planning scene

The operator, with the real-time update, is aware if the area around the robot is free of objects as it should be, as is the robot itself, since the planning scene in MoveIt! (Coleman et al., 2014), which already contains the static environment, is also updated as the mesh creation is combined with the function of OctoMaps (Hornung et al., 2013). MoveIt! is a widely used software for manipulation of robots, providing planners for the resolution of kinematics. An OctoMap, short for Octree-based 3D mapping, is a probabilistic mapping framework used in robotics and computer vision to represent the environment in three dimensions. It is particularly popular in scenarios where accurate spatial representation is crucial, such as autonomous navigation for drones or robots. Therefore, the filtered point cloud that creates the mesh is also generated in the planning scene as an obstacle, ensuring that there will be no collisions for the robot, ensuring dynamic obstacle avoidance. Additionally, the operator knows where the targets of the operation are situated, and thus they are able to move the robot to the desired position.

4.2.3 | Controlling the robot

The nodes for robot movement that are used for the remote control of the robot, work with the server–client implementation. The ROS system runs the server of the Cartesian movement, awaiting the client to send a movement command. In the proposed methodology, three different methods for remote control are presented: the end effector method, the joint values method, and the pose tracking technique.

In the case of the end effector-based movement, in order to give a command, the operator moves the virtual end effector of the robot within the VR environment, which represents a peripheral device that attaches to a robot's wrist, namely, a grasping tool. The virtual end effector is considered the desired end point of the real end effector. Using the VR interaction the operator designates a specific position with this method and sends the appropriate signal to MoveIt! motion planner. MoveIt! initially uses motion planners that calculate inverse kinematics to generate paths, which have no timing information associated with them and then uses trajectory processing algorithms that generate trajectories by working on these paths and are properly time—parametrized according to limits that are given to the joints of the robot. These planners calculate the trajectory toward the designated position using inverse kinematics algorithms. After the calculation of the plan, the operator can interact with a specific button in the VR interface to send the signal and execute the movement. Before the actual execution, the operator can command a visualization of the impending execution in VR. ROS publishes the calculated plan as a trajectory message that contains trajectory points, which in turn contain the positions, velocities, accelerations, and effort of the trajectory and can be transferred to VR and via ROSbridge. A copy of the robot model that is semi-transparent, called the "ghost robot," subscribes to and visualizes this trajectory. This works as a failsafe in case the planning scene is not fully updated and adds another layer of security to robot movements. The operator, through the created mesh and the other environmental awareness methods, can see that the trajectory passes through an object, which means collision, or if the movement is not satisfactory in any way. For example, the robot joints may turn too many times, which is not good for the cabling.

Aside from using the Cartesian commands to manipulate the robot, another method that was developed is the creation of a virtual controller that simulates the teach pendant of the robot, executing a linear movement of the end effector by using MoveIt! servoing capabilities, which allow the stream of end effector velocity commands to the manipulator and have it execute them concurrently. From the VR side, the operator can use a panel that has buttons that send velocity commands for each translation and rotation direction separately, in a very similar way as the end effector movement from a teach pendant.

The third remote robot control option is the joint values method. In the VR interface, the joint states of the robot are visualized in the format of decimal numbers that correspond to the rotation in radians of each joint of the robot. The operator has the option to interact with arrows and increase or decrease the numbers of the joint states by a specific increment, which can be adjusted using a virtual slider. For each increment, a signal is published directly to the robot controller through the manipulator topic. The controller updates the joint angles in real time according to the operator's interaction. The pose tracking technique and the joint values method both bypass MoveIt! sending commands directly to the robot's controller. Figure 6a depicts the three different methods for remote robot control from within the VR teleoperation application.

4.3 | Remote ROS communication

ROS is by its nature a framework dependent on local networks to function. In a usual setup, the robot, the PCs, and sensors are all connected in a local network. For teleoperation, a remote connection must be created between the mobile robot platform and the VR operator. Two methods were considered, VPN and port forwarding (Hajjaj and Sahari, 2017). They each have advantages and disadvantages over one another.

1. VPN is more secure than port forwarding as the traffic between client and server is encrypted, while port forwarding forward specific ports without any encryption.
2. VPN is more complex to set up, as it requires server and client settings setup, while port forwarding only requires router settings to be changed.
3. VPN is more flexible as it can provide access to multiple devices, while port forwarding is limited to a single device.
4. Port forwarding does not have any impact on network performance, while VPN does due to the encryption and decryption process.

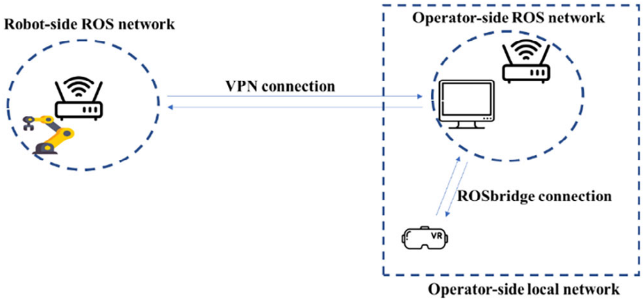

FIGURE 5 Remote connection concept.

The two methods were considered, and VPN was chosen as the better way due to safety and flexibility. For this reason, a peer-to-peer (P2P) VPN connection is proposed, bridging the two remote ROS networks, one on the local side of the operator and one on the remote robot side. Using this method, the device can make P2P connections with other devices in remote networks. Due to using a P2P connection, the data exchanged between the remote devices are sent directly with no central server in between, and everything is encrypted.

As shown in Figure 5, the remote network on the road will be connected via VPN to the local network of the operator. The ROS network can be considered a sub-network of this local network, containing a ROS-connected PC, which will receive the topics from the remote ROS network, and via ROSbridge will transfer them. This is a two-way connection since the VR system can send messages and goals to ROS through ROSbridge. The drawback to this method is the latency introduced when the Internet connection of the remote system is not fast enough. It is a drawback that is unavoidable, since the communication is a web-based system, and is countered with good connections, such as 5G systems.

4.4 | ROS interfaces

As mentioned, the communication between the VR application and the robotic system, which includes the robot, the cameras, and any other equipment used in the specific operation, is performed through an ROS connection. The ROS framework supports different ROS interfaces to facilitate the transmission of the necessary data. Table 1 presents examples of ROS interfaces, necessary for the establishment of the proposed methodology.

5 | CASE STUDY AND RESULTS

In order to prove the validity of the proposed concept, it was decided to apply the method in road maintenance interventions since they are the ideal example of an operation where teleoperation and training can be applied to great effect. The teleoperation will work in combination with a mobile robotic shell, consisting of a truck mounted with a robot. The robot can assist with maintenance interventions, taking the place of some, or all, operators. The deployment of the technology on the road will be reported in future publications. Most of the experiments shown below are in lab environment, although the same principles, as they regard the method, stand.

5.1 | Teleoperation

Although the teleoperation has not yet been tested on the road, which will be future work, in-lab tests have been undertaken to prove that the concept is sound. This means that the proposed method can be used in a practical application for the remote operator to teleoperate a remote robot successfully while being sufficiently aware of the remote environment to perform the operation safely. To begin with, an interface was created for the teleoperation application to assist the remote operator in choosing between options for the creation of the dynamic environment. The VR interfaces in the described method were created to be user-friendly, in an iterative process that took into account the opinions of approximately 100 users, during an exhibition in the TRA Lisbon 2022 conference, where a simulation of robot control in road maintenance was presented as a demo for the visitor to try. First, the operator enters a virtual lobby, depicted in Figure 6b.

Then the operator can choose to create an environment from three choices: A static VE with the truck and robot, an environment from Google Images, as shown in previous sections, and an environment updated by meshes. These environments can also be combined. Figure 6c illustrates the instructions to establish each VE.

For the skysphere environment that utilizes Google Images, the operator must input the location in coordinates. Then the skysphere is spawned around them and the static models, such as the truck and the robot, as shown in Figure 6d.

The mesh creation development was tested in a lab environment. An empty scene was loaded in Unity3D, containing only a UR10e robot that was connected to a real UR10e robot in the lab. The robot has a mounted camera, a D435 Realsense and “swept” the area around the robot, taking “snapshots” of point clouds at specific views, creating the immediate environment of a conveyor, and an operational table made from aluminum profiles. In Figure 7a,b, the comparison between the environment and the VE that was created is shown.

TABLE 1 General robot operating system (ROS) interfaces.

ROS interface	Description
sensor_msgs/JointState.msg	The virtual reality (VR) app receives the join states of the robot in this message type
trajectory_msgs/JointTrajectory.msg	The VR app publishes this message to move the robot by changing the joint states, as an alternative method of moving than the motion action server
geometry_msgs/PoseStamped.msg	The VR app sends an action goal that initiates the motion plan from MoveIt!
moveit_msgs/DisplayTrajectory.msg	The VR app receives a message of this type to visualize the plan that MoveIt! creates
sensor_msgs/PointCloud2.msg	The VR app receives a message of this type to generate the mesh out of the camera's point cloud
sensor_msgs/CompressedImage.msg	The VR app receives the compressed image of the camera using a message of this type

Control the robot
Move the robot before the signal. The camera must view the whole sign.

Street View Environment:
In the main menu, press 'Set Location'. Insert the coordinates of the location you want and press 'Submit Values'. Return to the main menu and press 'Start Operation'. Toggle on the 'Street View Environment' choice and press 'Start'.

FIGURE 6 VR interfaces: (a) robot teleoperation methods, (b) application virtual lobby, (c) application instructions, and (d) Google Street View images.

As evidenced, the environment in the immediate vicinity of the robot that was in the fields of view that the robot captured is faithful, although it contains noise. The quality of the output is reliant on the camera that is used. Realsense point clouds contain a lot of noise, even when the parameters are optimized. The algorithm that reduces the noise may overcompensate and remove details that are useful off the mesh, causing the scanning to take more time to create a complete environment. Tests done with better quality cameras such as Zed mini show an improved created mesh as shown in Figure 4e. Therefore, the issue of an incomplete mesh can be solved optimally with higher-quality cameras or with spending more time in re-scanning the area with a lower-quality camera. The speed of the

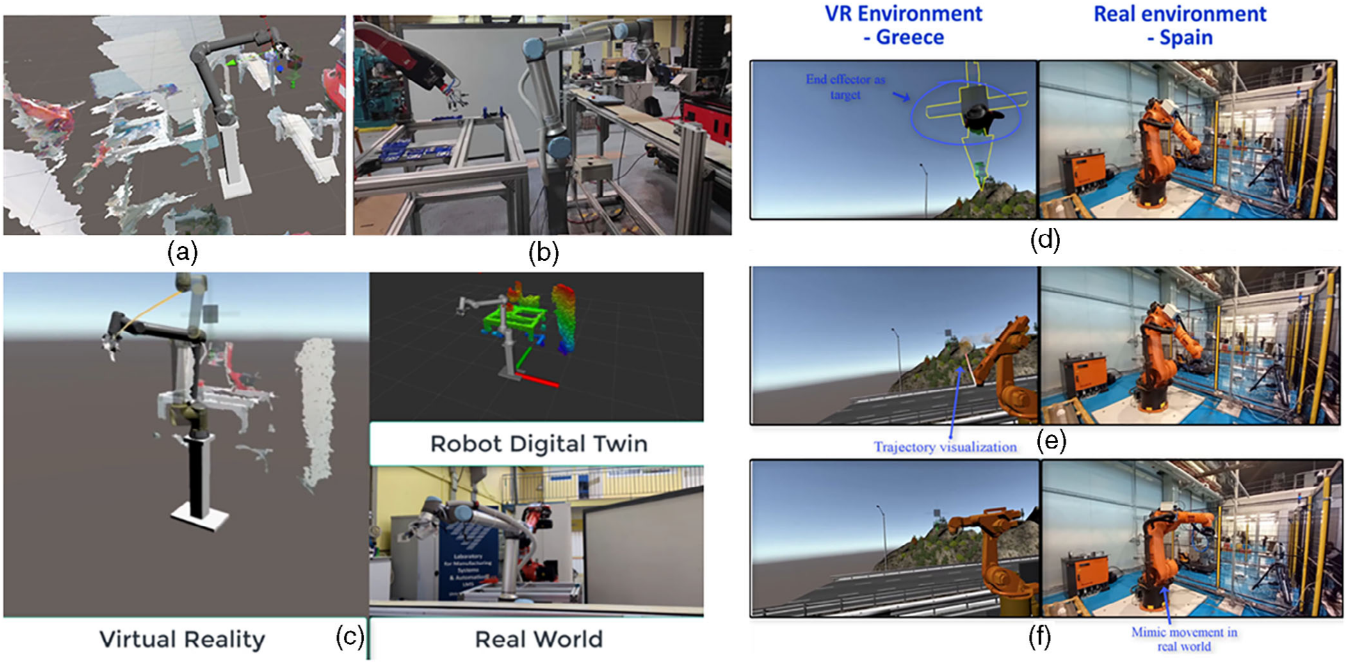

FIGURE 7 Robot teleoperation through VR application. (a) Reconstructed environment, (b) real environment of UR10e testbed, (c) robot teleoperation test (UR10e), and (d, e, f) robot teleoperation test (KUKA KR60).

reconstruction varies according to the speed of the Internet on both sides. In lab conditions, the speed is fast (80–100 Mbps), providing the reported results. On outdoor environments, the connection depends on the 5G network coverage, which may vary. While compressed images can easily be transported with 30 fps even in low speeds, point clouds are much worse, reaching as low as 2 fps. This is countered by filtering the point cloud to change from 300,000 points to 30,000, 10% of the initial cloud. Reducing the cloud any more creates problems with the meshes and makes the point cloud sparser, which lacks sufficient data points to accurately represent the surface geometry of the objects in the environment. Furthermore, irregularities in the distribution of points may complicate the connection of points to form a continuous surface, leading to inaccuracies in the mesh. The creation of meshes from sparse point clouds is a challenge that has been the subject of research in computer vision (Daroya et al., 2020).

The remote control of the robot was also tested on the UR10e robot. The VR application was connected via Rosbridge to a ROS network that ran in an Ubuntu virtual machine in the same personal computer (PC) as Unity, which runs on Windows. Then the ROS network in the virtual machine was connected with the robot's ROS network via VPN, according to the schema in Figure 5. In this test, the operator was controlling the robot using the end effector method as described in Section 4.2.3. Figure 7c depicts an example of this experiment. The operator, wearing the headset, moves the end effector of the virtual robot. The virtual end effector is visualized with the semi-transparent box that is shown in the left side of the figure. This virtual box has also been enhanced with a virtual axis, to help the operator identify the rotation of the virtual end effector. By interacting with this tool, the operator designates the position and rotation of the destination point for the movement of the real end effector of the robot. Then the operator interacts with the VR interface to order the planning of a trajectory. Following this interaction, the VR application sends a signal with the information of the destination point to the MoveIt! framework. This framework generates a trajectory toward that position and publishes this planned path through ROS. The VR application receives this information and visualizes the movement using a hologram of the robot, which is usually called a "ghost" robot due to its transparent material. This visualization is depicted on the left side of Figure 7c. If the operator is satisfied, they can interact with the VR interface again, to send the appropriate signal to the robotic system and execute the movement. While the real robot performs this movement, the operator can observe the virtual clone of the robot following the same exact trajectory within the VE. For the execution of this experiment, the two networks were in different buildings, with the distance being more than a 100 m, although the distance is considered irrelevant. In Figure 7, three views are presented. VR view, real-world view, and robot digital twin view in Rviz. VR view is the view of the remote operator, while DT view illustrates how the robot understands the environment around it. In this case, the point was to move the robot while being aware of the table in front of it to avoid collisions. In order to do that, the mesh of the table was created as explained above, in the VR application. The outcome of the point cloud processing that was performed for the mesh creation is also used in ROS, to generate the OctoMap of the surroundings of the robot in Rviz. This allows for dynamic obstacle inclusion in the planning scene of the robot. Thus, the operator is aware of the table because they can see it in VR and the robot planners understand the table as an obstacle to be avoided. This way, the operator can position the virtual end effector in the proper position that is not inside an object, and the MoveIt! planners take into account the objects when generating trajectories avoiding any collisions. The whole process of the experiment was repeated successfully several times.

The test was also performed for a much greater distance, from Greece to Spain. The connection was the same as the above test, with the VR and the virtual machine being in Greece, in the city of Patras, and the robot being in Spain, in Eibar. This time the robot was the KUKA KR60. The software used was the same with very little differences, confirming not only that the distance is irrelevant but also that the teleoperation application is robot-model-agnostic. The commands from the virtual robot to the real robot are much faster than the image data transference even in slow Internet speeds, although with a bad network, the update of the virtual robot joint positions when the real robot moves may be delayed for a few seconds in very bad connection, but that affects only visualization. Figure 7d–f presents examples of the performed robot teleoperation test using the KUKA KR60 robot.

Aside from robot control, this setup works to bridge VR with the remote ROS system and can be used for a variety of reasons, as it gives the ability to send commands from VR to ROS using services and publishers. As such, the remote operator can control any ROS-based function of the operation from afar, such as closing or opening a gripper, or commanding a detection from a camera. The topics utilized specifically for the remote teleoperation are shown in Table 2. As seen in Table 3, bad latency can lead to a delay in the generation of the 3D mesh, which negatively affects the life cycle.

To further test the efficiency of the developed implementation, another experiment was performed. This experiment aimed to assess the implementation in terms of network latency and determine how impactful this variable is for the efficiency of the proposed solution. Initially, to test the implementation under different network conditions, a specific tool that provides network emulation had to be installed on the remote PC device present at the activity area and connected to the robot. The tool chosen for this purpose was NetEm. NetEm is a powerful tool for Linux that enables users to introduce delays and packet loss (Hemminger et al., 2005).

By utilizing NetEm, a test was performed at three different latency levels. The initial latency of the remote network was 9–10 ms. Using NetEm, specific amounts of 10, 30, and 50 ms were added to the existing latency, creating three levels for the experiment: ∼20, ∼40, and ∼60 ms.

For each latency level, two different tests were performed. The first one involved the teleoperation of the robot. The operator designated a position to move the robot as described in Section 4.3. The duration between sending the signal from the VR tool and initiating the movement of the actual robot was measured for each latency level.

The second test involved the environment reconstruction process to create the dynamic environment using the point cloud from a camera mounted on the robot as described in Section 4.1. For each latency level, the duration of the entire mesh creation process was measured as described earlier in this section.

5.2 | Discussion of the results

We can separate the teleoperation results into three categories, dynamic environment update, remote communication, and robot control. As regards the Google Images skysphere for the dynamic update of the VE, an important thing to consider is that they are not updated daily, so they do not necessarily provide the latest outdoors environment. They also do not provide information on dynamic obstacles. But they are not intended to provide the latest update of the remote environment, rather they are intended to provide to the operator a general picture of how the environment is at the point of the intervention and to be combined with the other methods of dynamic update such as the mesh creation, and in that they work as they should. Mesh creation is meant to reconstruct the immediate environment of the robot as it is just before the execution of the process. Therefore, the operator is aware of the topology around the robot and any obstacles or changes not shown in the skysphere. While that would be enough for an environment that would stay stable at this time, there may be changes that happen during the execution. In order to account for these changes, and to assist in the execution, 2D image streaming is used from on-board cameras. The cameras can be on the robot and/or on the truck (or other parts of the static environment in other cases), providing views from appropriate angles. The results were satisfactory, in that the environment in the immediate vicinity of the robot is recreated in VR in sufficient enough detail that safe teleoperation is guaranteed. The recreation can be further improved by changing the camera into a better one without noise, which also means

TABLE 2 ROS interfaces for the communication of VR application.

ROS interfaces	Description
/joint_states	The VR app receives the join states of the robot from this topic
/manipulator_controller/command	The VR app publishes to this topic to move the robot by changing the joint states, as an alternative method of moving than the motion action server
/move_topose_action_server/arm/action	The VR app client sends an action goal, to the "move to pose" server that initiates the motion plan from MoveIt!
/retry_plan_topic	The VR app publishes to this topic when the operator wishes to replan and create a new trajectory of the robot
/execution_topic	The VR app publishes to this topic when the operator is satisfied with the plan and wishes to continue with the execution
/move_group/display_planned_path	The VR app subscribes to this topic to visualize the plan that MoveIt! creates
/triangles	The VR app subscribes to this topic to receive the triangles and vertices for the mesh creation
/camera/color/image_raw/compressed	The VR app subscribes to this type of topic to receive images from ROS-connected sensors
/camera/color/image_raw/compressed	The VR app subscribes to this type of topic to receive images from ROS-connected sensors

TABLE 3 Robot teleoperation and mesh creation duration for different latency levels.

Latency (ms)	Robot teleoperation duration	Mesh creation duration
20	~0.1 s	20–25 s
40	~0.2 s	40–50 s
60	~0.4–0.6 s	2.5–3 min

changing the parameters of the filtering of the point cloud. These methods are combined with images from the on-board cameras on the robot platform, to complete the image of the remote environment for the remote operator.

The control of the robot by using the application is easy to use even for operators who are not experts in robotics. The movement by the end effector enables the operator to move the robot to an approximate desired position, depending on the accuracy of the operator's hand. For more accuracy, the virtual controller offers the capability of moving the robot with accuracy to the joint value, which is necessary for smaller, more precise movements, and for even more accuracy, the operator can use linear movement of the end effector. There is no need for the operator to be educated in robotics, as they do not need to use a controller, or code for the movements, nor calculate the kinematics themselves. In Figure 8, a qualitative calculation of efficiency and safety when the operator is in the operation environment in comparison with the proposed teleoperation method is shown.

Old-fashioned operations, where everything is done manually by experienced operators, are characterized by efficiency. The largest concern is safety and ergonomics. Safety measures are of course taken, although they are limited. For example, in road maintenance, one lane, the lane where the intervention is executed, is usually closed for the safety of the operators, but the traffic in other lanes continues, with even more packed traffic, since there are now fewer lanes. There are cases where cars have accidentally driven in the closed-off lane and caused deadly accidents for the operators. In other sections, such as manufacturing, where robots and humans work in the same shell, the operations are quite dangerous even with cobots, and many accidents have been reported. Other operations may contain other hazards, such as inhalations of harmful substances. Additionally, the loads the operators must carry during the operation are often quite heavy and tiring and can often lead to musculoskeletal issues.

The VR teleoperation application is focused mostly on the safety of the operators. By removing the operator from the field, it is guaranteed that work accidents—as they regard the hazardous environment—will not happen to them, reducing the risk to the possibility of the residual risks in the safe place the operator works in. Efficiency is hurt, by possible latency as well as more processes that are now necessary to be executed, such as the recreation of the environment. As seen in Table, bad latency can lead to a delay in the generation of the 3D mesh, which negatively affects the life cycle. Latency is a natural phenomenon in web-based communication and is dependent both on the strength of the network, as well as the communication method. The proposed method suggests VPN since it is a secure method of communication that offers good latency levels when the network is fast, but there are other methods, such as port forwarding. A comparative analysis of the different methods and their benefits will be the subject of future work. User-friendly robot control interfaces raise efficiency since they enable operators who are not experienced with robotics to control a robot during

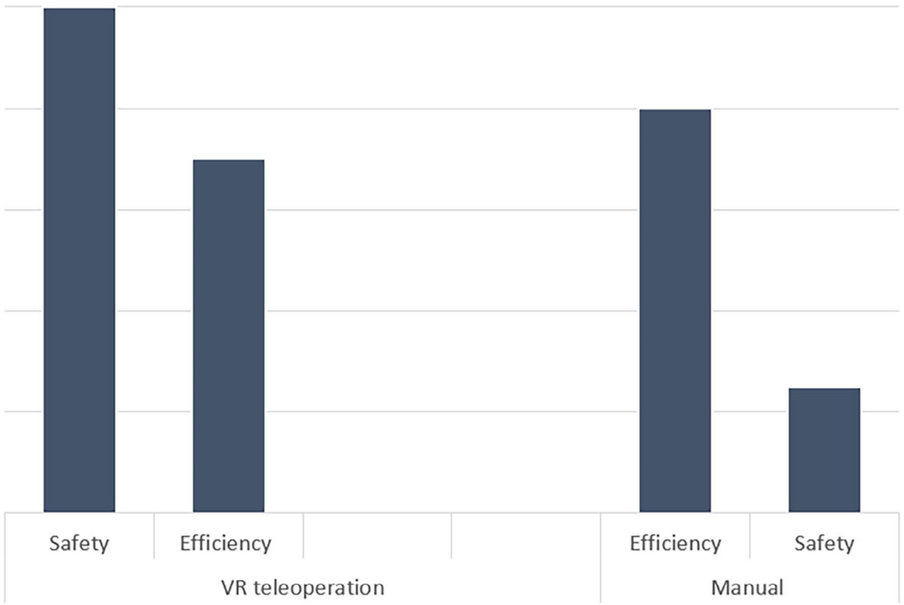

FIGURE 8 Qualitative chart of targets.

execution. When approximately 100 participants tried the application in the TRA Lisbon 2022 conference, the time it took to familiarize with the controls was no more than 5 min on average. This introduces the advantages of the robot in handling operations, keeping efficiency to a satisfactory level. Teleoperation is based on the robot doing the heavy lifting while the operator is only using VR, therefore the operator does not get tired physically.

On the topic of scalability, as was shown, the method can be used with different robots, as long as they can connect to ROS. If there are no ROS drivers ready for a specific robot, they can be developed. Different applications may require different handling operations or different machines, and therefore different ROS messages will need to be created according to the needs of the operation. Of course, the proposed method, as mentioned in the introduction, works for industrial operations in which the robot can handle all operations with the remote control of the human, and therefore a challenge would be to adapt the operations to this model. Regarding more complex environments, and specifically obstacles, the method already adapts due to the dynamic VE recreation. The challenge in these environments is to arrange the cameras in a way that provides the right angles and an appropriate point cloud for the operator to have full awareness. Also, in the case of indoors environments, the Google Images environment will not work, although skyboxes and skyspheres can be created with any kind of continuous images. As it regards the differences between hazards for the human between different industries, no specific adaptation is necessary, as the human is removed from the equation according to the proposed method. Regarding the socioeconomic impact, this method can have in the industry, it speaks to the reason that there will be reduced healthcare costs due to the human being in a safe environment, as well as reduced labor costs due to higher automation. The adoption of advanced technology such as VR teleoperation can also drive innovation in industry. Also, upskilling workers will be necessary to a degree, in order for them to be able to use VR effectively. Also, since the method allows the operator to work from anywhere in the world if they have an Internet connection, geographic flexibility is enabled for companies.

6 | CONCLUSION

In this manuscript, a method for the enhancement of human–robot collaboration in hazardous operations, based on VR technology is proposed, specifically a VR teleoperation tool for remote robot and operation control and monitoring. The teleoperation tool is based on the dynamic update of the VE, essentially the creation of a digital twin of the remote environment. The dynamic update is based on the creation of a 3D mesh of the surrounding area of the robot, as well as camera streams and skysphere created using Google Images. The remote connection is web-based and handled by a VPN P2P connection, and three distinct methods for robot control were developed and tested. This paper attempts to explain the methods and provide proof of concept of these tools in road maintenance use cases, which serve as an example of hazardous operations. The enhanced features that the developed tools offer to the road operator are the following:

1. Efficiency. User-friendly interfaces allow for human–robot collaboration with efficiency, reducing the physical load for the operators.
2. Safety. The impact of the proposed method is greater on safety. By using teleoperation, the operator will no longer need to be on the road, in dangerous environments, and will be able to monitor and execute the intervention from home or office, eliminating the possibility of any sort of industrial accident. In terms of statistics, according to Eurostat, there is a 15.4% percentage of non-fatal accidents in 2021, which happened in tertiary sites, such as offices. In comparison, in 2021, more than 22.5% of all fatal accidents happened within the construction sector. The tertiary sites such as offices are not even mentioned as a percentage of fatal accidents. Therefore, the proposed method, which removes the operator from the construction site and moves them to an office, has a staggering effect on safety.

A limitation of the proposed method is the lack of specific strategies for the mitigation of data loss and delays during the remote transmission of data, aside from the VPN properties, being dependent on the strength of the network. Succeeding over lossy communication networks has been active research in telerobotics for quite some time as reported in Siciliano et al. (2008). The development of said strategies could be part of future work. Another limitation is in the dependence on the current stage of technological advancement for Google Images skysphere, which is constantly updated and therefore not used to the maximum potential an up-to-date skysphere environment could have. The quality of the sensors used can also be considered a limitation. The choice of a camera has an important impact on the quality of the mesh that is created. A higher-quality 3D camera will create a higher-quality 3D mesh in much less time than a lower-quality camera, which may need a longer sweep of the area for a functioning mesh to be generated. Finally, this method works as long as the hazardous environment allows 3D sensors to capture data.

The developments that were analyzed here will be further validated in future work, in a more quantitative manner, where they will be tested on demonstrators on actual road environments. Future work will contain the integration of the teleoperation platform with a modular robot platform, consisting of a truck, mounted with a robot and containing the necessary road assets for the interventions.

ACKNOWLEDGMENTS

This work has been partially funded by the EC 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

Arevalo Arboleda, S., Rücker, F., Dierks, T., & Gerken, J. (2021). Assisting manipulation and grasping in robot teleoperation with augmented reality visual cues. *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, Yokohama, Japan (pp. 1–14).

Balta, H., Velagic, J., Bosschaerts, W., De Cubber, G., & Siciliano, B. (2018). Fast statistical outlier removal based method for large 3D point clouds of outdoor environments. *IFAC-PapersOnLine*, *51*(22), 348–353.

Barkokebas, R., Ritter, C., Sirbu, V., Li, X., & Al-Hussein, M. (2019). Application of virtual reality in task training in the construction manufacturing industry. *ISARC. Proceedings of the International Symposium on Automation and Robotics in Construction*, Banff, Canada (Vol. *36*, pp. 796–803).

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

Coleman, D., Sucan, I., Chitta, S., & Correll, N. (2014). *Reducing the barrier to entry of complex robotic software: A MoveIt! case study*. arXiv preprint arXiv:1404.3785. https://doi.org/10.48550/ARXIV.1404.3785

Cutini, M., Bisaglia, C., Brambilla, M., Bragaglio, A., Pallottino, F., Assirelli, A., & Menesatti, P. (2023). A co-simulation virtual reality machinery simulator for advanced precision agriculture applications. *Agriculture*, *13*(8), 1603.

Daroya, R., Atienza, R., & Cajote, R. (2020). REIN: Flexible mesh generation from point clouds. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, Seattle, WA (pp. 352–353).

Dias, P., Matos, M., & Santos, V. (2006). 3D reconstruction of real world scenes using a low-cost 3D range scanner. *Computer-Aided Civil and Infrastructure Engineering*, *21*(7), 486–497.

Dimitropoulos, N., Togias, T., Michalos, G., & Makris, S. (2020). Framework enabling the design of Virtual Environments used for simulation of assembly operations. *Procedia Manufacturing*, *51*, 571–576.

Eskandari Torbaghan, M., Kaddouh, B., Abdellatif, M., Metje, N., Liu, J., Jackson, R., & 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.

Fishel, J. A., Oliver, T., Eichermueller, M., Barbieri, G., Fowler, E., Hartikainen, T., Moss, L., & Walker, R. (2020). Tactile telerobots for dull, dirty, dangerous, and inaccessible tasks. *2020 IEEE International Conference on Robotics and Automation (ICRA)*. https://doi.org/10.1109/icra40945.2020.9196888

Hajjaj, S. S. H., & Sahari, K. S. M. (2017). Establishing remote networks for ROS applications via Port Forwarding: A detailed tutorial. *International Journal of Advanced Robotic Systems*, *14*(3), 1729881417703355.

Hemminger, S. (2005). Network emulation with NetEm. *Proceedings of Australia's 6th National Linux Conference*, Canberra, Australia.

Hornung, A., Wurm, K. M., Bennewitz, M., Stachniss, C., & Burgard, W. (2013). OctoMap: An efficient probabilistic 3D mapping framework based on octrees. *Autonomous Robots*, *34*, 189–206.

Kamran-Pishhesari, A., Moniri-Morad, A., & Sattarvand, J. (2024). Applications of 3D reconstruction in virtual reality-based teleoperation: A review in the mining industry. *Technologies*, *12*(3), 40.

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* (pp. 1–4). IEEE.

Katsamenis, I., Bimpas, M., Protopapadakis, E., Zafeiropoulos, C., Kalogeras, D., Doulamis, A., & Lopez, R. (2022). Robotic maintenance of road infrastructures: The heron project. *Proceedings of the 15th International Conference on PErvasive Technologies Related to Assistive Environments*, Corfu, Greece (pp. 628–635).

Kim, N., Yan, N., Grégoire, L., Anderson, B. A., & Ahn, C. R. (2023). Road construction workers' boredom susceptibility, habituation to warning alarms, and accident proneness: Virtual reality experiment. *Journal of Construction Engineering and Management*, *149*(2), 04022175.

Lanz, M. (2022). Feasibility analysis of safety training in human–robot collaboration scenario: Virtual reality use case. In K. Y. Kim, L. Monplaisir, & J. Rickli (Eds.), *Lecture notes in mechanical engineering. Flexible automation and intelligent manufacturing: The human—data—technology nexus. FAIM 2022* (pp. 246–256). Springer.

Li, D., Liu, J., Feng, L., Zhou, Y., Qi, H., & Chen, Y. F. (2021). Automatic modeling of prefabricated components with laser-scanned data for virtual trial assembly. *Computer-Aided Civil and Infrastructure Engineering*, *36*(4), 453–471.

Luleci, F., Li, L., Chi, J., Reiners, D., Cruz-Neira, C., & Catbas, F. N. (2022). Structural health monitoring of a foot bridge in virtual reality environment. *Procedia Structural Integrity*, *37*, 65–72.

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

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.

Miknis, M., Davies, R., Plassmann, P., & Ware, A. (2015). Near real-time point cloud processing using the PCL. In P. Liatsis (Ed.), *2015 international conference on systems, signals and image processing (IWSSIP)* (pp. 153–156). IEEE.

Miyamoto, A. (2021). Development of a remote monitoring system for road condition assessment and application. In D. M. Frangopol & H. Yokota (Eds.), *Bridge maintenance, safety, management, life-cycle sustainability and innovations* (pp. 371–380). CRC Press.

Pan, Y., Chen, C., Li, D., Zhao, Z., & Hong, J. (2021). Augmented reality-based robot teleoperation system using RGB-D imaging and attitude teaching device. *Robotics and Computer-Integrated Manufacturing*, *71*, 102167.

Peng, S., Cheng, X., Yu, M., Feng, X., Geng, X., Zhao, S., & Wang, P. (2023). Collision risk assessment and operation assistant strategy for teleoperation system. *Applied Sciences*, *13*(7), 4109.

Pérez, L., Diez, E., Usamentiaga, R., & García, D. F. (2019). Industrial robot control and operator training using virtual reality interfaces. *Computers in Industry*, *109*, 114–120.

Shneiderman, B., & Plaisant, C. (2010). *Designing the user interface: Strategies for effective human-computer interaction*. Pearson Education India.

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)*, Online.

Siciliano, B., Khatib, O., & Kröger, T. (Eds.). (2008). *Springer handbook of robotics* (Vol. 200). Springer.

Solanes, J. E., Muñoz, A., Gracia, L., & Tornero, J. (2022). Virtual reality-based interface for advanced assisted mobile robot teleoperation. *Applied Sciences*, *12*(12), 6071.

Togias, T., Gkournelos, C., Angelakis, P., Michalos, G., & Makris, S. (2021). Virtual reality environment for industrial robot control and path design. *Procedia CIRP*, *100*, 133–138.

Zhang, J., Yu, F., Li, D., & Hu, Z. (2014). Development and implementation of an industry foundation classes-based graphic information model for virtual construction. *Computer-Aided Civil and Infrastructure Engineering*, *29*(1), 60–74.

Zhang, Z., Wen, F., Sun, Z., Guo, X., He, T., & Lee, C. (2022). Artificial intelligence-enabled sensing technologies in the 5G/Internet of Things era: From virtual reality/augmented reality to the digital twin. *Advanced Intelligent Systems*, *4*(7), 2100228.

Zhao, S. N., Cui, Y., He, Y., He, Z., Diao, Z., Peng, F., & Cheng, C. (2023). Teleoperation control of a wheeled mobile robot based on Brain-machine Interface. *Mathematical Biosciences and Engineering*, *20*(2), 3638–3660.

Zhou, T., Zhu, Q., & Du, J. (2020). Intuitive robot teleoperation for civil engineering operations with virtual reality and deep learning scene reconstruction. *Advanced Engineering Informatics*, *46*, 101170.

Zhu, Q., Zhou, T., & Du, J. (2022). Upper-body haptic system for snake robot teleoperation in pipelines. *Advanced Engineering Informatics*, *51*, 101532.

How to cite this article: Bavelos, A. C., Anastasiou, E., Dimitropoulos, N., Michalos, G., & Makris, S. (2024). Virtual reality-based dynamic scene recreation and robot teleoperation for hazardous environments. *Computer-Aided Civil and Infrastructure Engineering*, 1–17. https://doi.org/10.1111/mice.13337