machines

MDPI

Article

Digital Twins Utilizing XR-Technology as Robotic Training Tools

Tero Kaarlela [1,*], Paulo Padrao [2], Tomi Pitkäaho [1], Sakari Pieskä [1] and Leonardo Bobadilla [2]

[1] Department of Industrial Management, Centria University of Applied Sciences, 84100 Ylivieska, Finland
[2] Knight Foundation School of Computing and Information Sciences, Florida International University, Miami, FL 33199, USA
* Correspondence: tero.kaarlela@centria.fi

Abstract: Digital technology has evolved towards a new way of processing information: web searches, social platforms, internet forums, and video games have substituted reading books and writing essays. Trainers and educators currently face the challenge of providing natural training and learning environments for digital-natives. In addition to physical spaces, effective training and education require virtual spaces. Digital twins enable trainees to interact with real hardware in virtual training environments. Interactive real-world elements are essential in the training of robot operators. A natural environment for the trainee supports an interesting learning experience while including enough professional substances. We present examples of how virtual environments utilizing digital twins and extended reality can be applied to enable natural and effective robotic training scenarios. Scenarios are validated using cross-platform client devices for extended reality implementations and safety training applications.

Keywords: digital twin; extended reality; robotics

check for updates

Citation: Kaarlela, T.; Padrao, P.; Pitkäaho, T.; Pieskä, S.; Bobadilla, L. Digital Twins Utilizing XR-Technology as Robotic Training Tools. *Machines* **2023**, *11*, 13. https://doi.org/10.3390/machines11010013

Academic Editor: Zheng Chen

Received: 28 November 2022
Revised: 12 December 2022
Accepted: 19 December 2022
Published: 22 December 2022

Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

The digital-native generation of today requires new ways of training and education [1–4]. Compared to previous generations, who grew up reading literature and comics, digital-natives have grown up surrounded by digital appliances, including cell phones, computers, and video games [5,6]. Growing up with the possibilities of new digital technology, a new way of processing information has evolved: instead of reading books and writing essays, digital-natives learn by utilizing web searches, social platforms, and internet forums. Digital natives prefer browsing graphical and video content over reading text documents; the new way of learning is more technological and digitally oriented. Trainers and educators of today are facing the challenge of providing training and learning environments natural for digital natives. In addition to physical spaces, effective training and education require cyberspaces [7]. An environment natural to the trainee supports the learning process, providing a positive learning experience and outcome. In this paper, virtual environments (VEs) utilizing extended reality (XR) are experimented on to enable natural and effective training environments for the digital native generation.

VEs are one of the most popular ways of interacting with each other for digital natives [8]. VEs enable simulations and demonstrations impossible to implement in a traditional classroom, bound to physical and geographical constraints. XR divides into sub-categories of augmented reality (AR), augmented virtuality (AV), and virtual reality (VR) [9]. AR, AV and VR enable the creation of immersive and augmented realistic training environments for the digital-native generation. AV and AR enrich virtual and real environments by adding real-world elements to the virtual world or virtual elements to the real world, creating mixed reality (MR) training experiences. VR enables an immersive training experience where the trainee immerses in a virtual environment. While XR currently applies to a wide variety of training applications, only a few aim for remote training of basic

robotic skills. Furthermore, extensive studying of AV implementations for robotic training is required. XR enables risk-free remote training for controlling, programming, and safety of industrial robots. The benefits of safety training in a simulated environment have been known for decades [10]. Morton Heilig stated in his patent application that the training of potentially hazardous situations and demanding industrial tasks should be carried out in a simulated environment rather than exposing trainees to hazards of the real environment.

A fully simulated training environment does not provide a realistic experience to the trainee; therefore, interfacing with real equipment is required [11,12]. If the trainee is allowed to control real equipment and see the results in a real environment, training experience is very close to hands-on experience. Digital twins (DTs) enable the user to interact with real hardware utilizing VEs, providing a bidirectional communication bridge between real and virtual worlds [13]. Real-time data such as speed, position, and pose from physical twins to digital twins instead of simulated data enable a realistic training experience. In addition, augmenting data of the surrounding environment to digital twin is possible [14,15]. Furthermore, DTs enable the validation of data such as robot trajectories before committing data to real hardware, enabling safe teleoperation of robots [16]. In the system proposed in this paper, DTs enable trainees to interact with real-world robot hardware. After validating the trajectories created by the user, DTs pass trajectories to the physical twin, and the physical twin provides feedback sensor data to DT. The proposed solution is intended as a virtual training environment, not as a robotics simulator such as Gazebo [17] or iGibson [18].

The industry is starting to implement the Metaverse, utilizing DTs as the core [19]. By creating DTs of physical equipment, virtual spaces, and data communication in between, the first steps towards the Metaverse are taken [20]. The Metaverse has the potential to merge physical reality and digital virtuality, enabling multi-user VEs. The latest implementations of the industrial Metaverse include training, engineering, working, and socializing [21].

In our previous work, we proposed a common teleoperation platform for robotics utilizing digital twins [22]. This paper extends our previous work by adding VEs to the proposed platform, enabling virtual training for basic robotic skills such as manual control and safety functionalities of industrial robots. The research question is: What are the benefits and challenges of DT and XR in remote robotic training? The contributions of this publication are as follows:

1. Three XR training scenarios for robotic safety, pick and place, and assembly.
2. Web-based XR architecture to support cross-platform functionality on both desktop and mobile devices.
3. DTs enabling teleoperation and trajectory validation of physical robots.

The paper is organized as follows: Section 2 explains methods and approach, including requirement specification for the proposed VEs. Section 3 provides a review of previous research of the topic. Section 4, describes the implementation and validation of proposed VEs. Section 5, provides further discussion of the results. Section 6 provides the conclusion of this publication.

2. Methods and Approach

This section presents methods utilized during the development of the proposed solution. The planning phase includes requirement specifications for the XR-training environment. Section 2.1 presents requirements specified for VEs, providing a fundamental guideline for the implementation described in the Section 4. The first task of the development phase is to define and validate the system architecture. After defining the architecture, the development continues with the implementation. The implementation consists of iterative modeling, programming, and functionality validation tasks. The validation task consists of three training scenarios to validate the functionality of VEs. The first training scenario studies the utilization of augmented virtuality in a traditional pick-and-place task. The second training scenario utilizes augmented reality to visualize industrial robot reachability, and the third scenario enables an immersive human–robot collaboration safety

training experience. After validating VEs against the requirement specification, VEs are considered ready for evaluation in real training applications. Figure 1 presents the methodology utilized in this work for planning, developing, and implementing the proposed VEs. Table 1 presents requirements specified for the VEs.

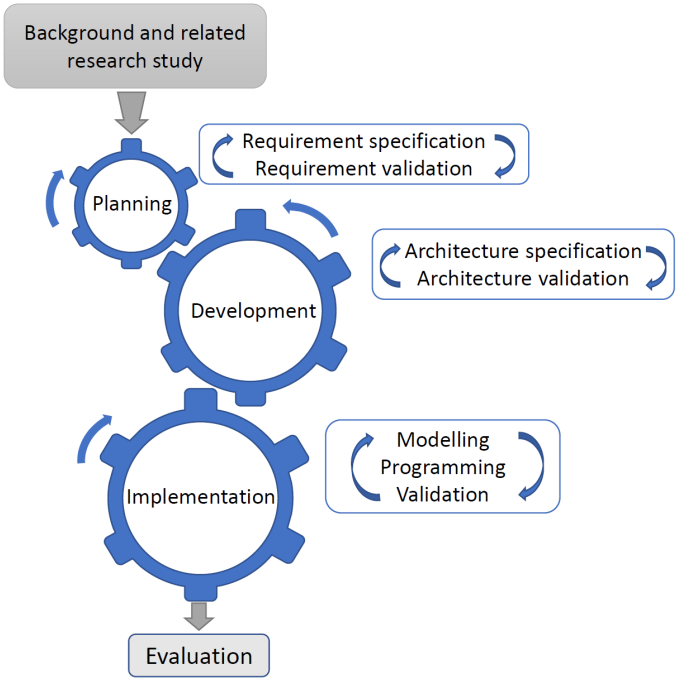

Figure 1. Methodology used in this work to develop and validate virtual environments. The foundation for planning is based on published results in the field and close collaboration with the industry. In the planning phase, requirements for the system are defined. Based on the specification, the system architecture is developed and validated. During the iterative implementation, the system is built and validated for the final evaluation.

Table 1. Requirements specified for virtual environments.

ID	Description	Category
N1	The systems clearly focuses on robotics.	Usability
N2	Platform independent.	Usability
N3	CS is considered. Remote users cannot access the system without authentication. Additionally, robot functionality is restricted for safe operations.	Security
N4	Safety taken into consideration in remote usage.	Security
N5	System updates and upgrades do not require actions from end-user.	Usability
N6	Delay between virtual and physical robots should be less than 250 ms.	Performance
N7	Exercises can be in a form of a 3D experience or desktop game.	Usability
N8	Specific VR-devices are not required	Usability
N9	User can teleoperate and program robots	Usability

2.1. Requirement Specification

The authors of this paper, having pedagogical, XR, robotics, and software development backgrounds, specified requirements for the VEs. The requirements specified are non-functional, defining quality-related requirements, such as how long the delay is between DT and the physical twin. The requirements were divided into sub-categories of usability, security, and performance, depending on the nature of the specific requirement. Security

requirements consider cybersecurity and physical security issues since a cybersecurity breach can compromise both. The requirements presented provide a fundamental guideline for the implementation phase.

3. Background and Related Research

The related research study aims to investigate and select hardware and software components required to build DTs utilizing XR technology as robotic training tools. The chosen components are required to meet requirements specified in Section 2.1. An overview of research on virtual reality and digital twins is provided, with a focus on the training of robotics. A study of XR hardware and software provides essential information on available tools to create and experience VEs and to select the correct tools for our implementation.

3.1. Extended Reality

Extended reality has been an active research topic for decades; in the early days, a common term was artificial reality instead of extended reality. The first steps towards providing a VR experience were taken in 1955 when Morton Heilig published his paper "The cinema of the future" describing a concept for a three-dimensional movie theater [23]. Later in 1962, Heilig presented The Sensorama, a device based on his concept providing the user with an immersive three-dimensional experience. Sensorama enabled the user to sit down and experience an immersive motorcycle ride through New York [24]. In addition to the visual experience, Sensorama provided the user with sensations of scents, vibration, wind, and audio [25]. Sensorama met all characteristics of virtual reality except one since the filmed route was static. Sensorama lacked interactivity, an essential element of virtual reality [26,27]. NASA's Virtual Visual Environment Display (VIVED), introduced in the early 1980s, was the first head-mounted display (HMD) documented to enable all characteristics of virtual reality: interaction, immersion, and imagination. The term virtual reality was introduced by Jaron Lanier later in the eighties [28]. Lanier's company VPL Research a spinoff from NASA, developed and marketed several XR devices including, EyePhone, Data Glove, and Data suit. EyePhone was one of the first commercially available VR headsets.

Augmented reality was first presented in 1968 by Ivan Sutherland as "A head-mounted three-dimensional display" [29]. While HMDs for applications such as video surveillance and watching TV broadcasts had been proposed earlier [30–32]. Sutherland's contribution was to replace external video feedback with computer-generated graphics, and the first interactive AR system was born. Due to limitations set by the computing power of the era, Sutherland's system was capable of providing only three-dimensional wireframe objects in front of the user's eyes. The term augmented reality was coined later by Tom Caudell in 1990 when he was investigating see-through HMDs for tasks related to aircraft manufacturing [33].

Myron Krueger [34] studied augmented virtuality in his project "responsive environments". Krueger's idea was to create a telepresence concept that would provide a virtual space where people distant from each other could interact in a shared virtual space. Krueger's Videoplace experiment consisted of rooms with a projector, a sensing floor, and a camera as a motion-detecting device to capture user movements and gestures. Videoplace enabled an augmented virtuality experience, augmenting the user into a virtually generated shared experience projected onto the walls of all connected rooms.

Virtual-Reality continuum, presented by Paul Milgram in 1994, defines XR as the main category for all technologies, including virtual objects [9]. Augmented reality and augmented virtuality form a subcategory of XR called Mixed Reality. Figure 2 clarifies XR and related subcategories.

AR, AV, and VR environments are implemented, presented, and discussed in this paper. Discussion Section 5 presents a comparison between entirely virtual and augmented training environments.

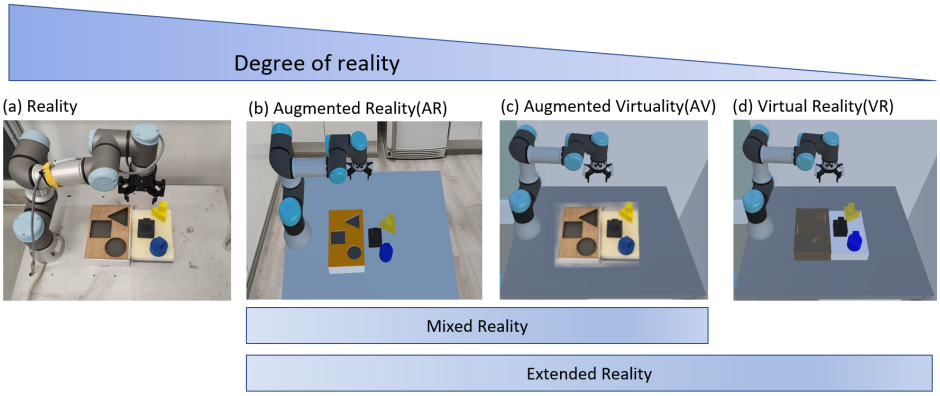

Figure 2. Reality–virtuality continuum. (**a**) real environment, (**b**) augmented virtuality, (**c**) augmented reality, and (**d**) virtual reality. The degree of reality decreases when moving from a real environment to virtual reality.

3.2. Extended Reality Hardware

Modern devices enabling XR-experiences are a result of long-term development work of researchers and innovators in the fields of entertainment [10], optics [35], electronics [36], and computing [29]. Immersive experience requires a minimum field of view (FOV), refresh rate, and resolution. Achievements in optics contribute to providing FOV wide enough to cover the entire area the user can see to provide an immersive virtual experience [26,35,37]. If the user can see a glimpse of the VR-headset frame instead of the virtual scene, the sense of immersion distorts immediately. Tracking the movement of the user's head and adapting perspective accordingly also contributes to creating a sense of immersion for the user [25]. User movement tracking systems can be either integrated or external to/of the headset [38,39]. Current external tracking systems enable twice as large a play area for the user compared to integrated tracking systems. Evolution in display technologies enables light, high-resolution display units capable of providing realistic graphics quality [24]. HMDs, now known as VR headsets, combine optics, displays, tracking devices, and speakers, into one product enabling an immersive virtual reality experience. VR headsets are capable of both standalone operation without a PC and operating as a connected accessory to a PC. Headsets connected to a PC can utilize high-end graphics processors and wired or wireless connections between PC and headsets [40]. Wireless VR headsets free the user from hanging cables, enabling the user to move freely within the limits of the play area. Current high-end systems such as Varjo XR-3 released in 2021 enable high-resolution wide field of view augmented, virtual and mixed-reality experiences [41]. The most affordable devices to experience extended reality are mobile phones and tablets. Mobile devices enable support for augmented reality by utilizing built-in position sensors and camera units. An immersive VR experience requires Google cardboard or similar VR-headset casing and the mobile device itself. A combination of mobile devices and Google Cardboard provides an affordable way to experience extended reality with the cost of lower FOV and resolution specifications compared to purpose-built devices. To ensure cross-device compatibility and to compare the significance of resolution, refresh rate, and FOV in training applications, VE implementations in this paper are validated utilizing Oculus Quest(Meta Platforms, Inc., Menlo Park, California, USA), HTC Vive(HTC Corporation, Taoyuan City, Taiwan) and Varjo XR-3 VR(Varjo Technologies Oy, Helsinki, Finland) headsets. A comparison in Table 2 presents the specifications of VR headsets. Specifications of the early EyePhone provide a baseline for comparison.

Table 2. Technical details of VR-headsets .

System	AR/VR	Year	Resolution	Weight	FOV	Tracking
EyePhone [24,26]	VR	1989	360 × 240	2400 g	90H × 60V	external
HTC Vive [39,42]	VR	2016	1080 × 1200	563 g	110H × 113V	external
Oculus Quest I [40]	VR	2019	1600 × 1440	571 g	115H × 90V	internal
Varjo XR-3 [41]	AR/VR	2021	2880 × 2770	980 g	115H × 90V	both

3.3. Web Based Extended Reality

Web-based XR applications are accessible, cybersecure, and not tied to specific hardware platforms or devices meeting non-functional requirements N2 and N3. Furthermore, the updating or upgrading of web-based applications does not require any user actions meeting requirement N5. For the aforementioned reasons, the web-based approach is chosen as the implementation for the VEs proposed. This section presents standards and protocols related to building web-based XR applications.

HTML5 is the latest version of HyperText Markup Language (HTML) released in 2008 [43]. HTML5 provides the user with advanced functions such as local file handling and support for complex two- and three-dimensional graphics [44]. HTML5 enables the embedding of three-dimensional graphics inside a Canvas element on any web page. Canvas application programming interface (API) provides a rectangular element as a graphics container on a web page. The graphic content creation inside the canvas element is enabled utilizing JavaScript. JavaScript is well known as a lightweight scripting language for web pages. JavaScript enables the real-time update of graphic content inside the Canvas element.

Web Graphics Library (WebGL) JavaScript API is capable of producing hardware-accelerated interactive two- and three-dimensional graphics inside the HTML5 canvas element. Together, WebGL and HTML5 canvas enable presenting three-dimensional interactive graphics embedded into a web page. Furthermore, the standards enable cross-platform XR-application development for desktop and mobile devices.

The WebXR device API (XRDA) standard enables a platform-independent interface to access hardware required for interactive XR-experience [45]. XRDA provides access to extended reality device functionalities such as the pose or position of the controller, headset, or camera and utilizes WebGL to render augmented or immersive virtual graphics accordingly. WebXR enables an extended reality experience on a desktop PC or a mobile web browser [46]. Figure 3 describes the structure of a WebXR-application.

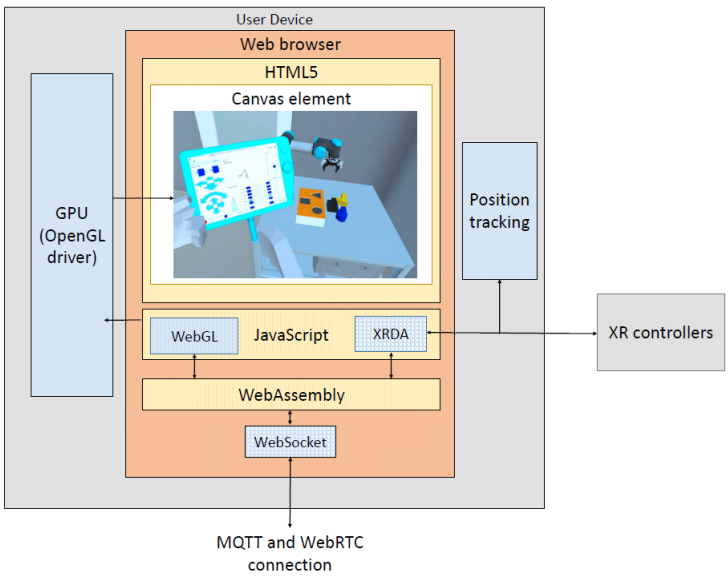

Figure 3. Structure of WebXR-application.

WebAssembly (Wasm) is a binary instruction format for a stack-based virtual machine. Four major browsers support Wasm to provide native speed execution of web-based applications [47]. During compilation, native formats such as C#, C++, and GDScript compile into Wasm format. As Wasm code does not have access to hardware, Javascript code is required to enable access to XR devices and graphics processing unit (GPU). The latest versions of the popular HyperText Transfer Protocol (HTTP) server Apache provides server-level support for Wasm [48].

3.4. Digital Twin Concept

Michael Grieves presented the concept of DT in 2002 as a conceptual idea for Product Lifecycle Management (PLM) [49]. The conceptual idea includes a virtual and real system linked together from the design phase to the disposal of the product. DT definition has been evolving since, and a virtual copy of a physical system enabling bidirectional interaction is one of the definitions [50]. Kritzinger defined automated bidirectional data flow as a unique feature of a DT, distinguishing DT from digital shadow and digital model [51]. Digital twin consortium defined data flow as a synchronized event at a specified frequency [52]. Automated data flow is an essential element of the DT concept, enabling interaction with the low-level functions of the physical twin and digital presentation of the physical twin information. DTs can present an individual system such as a robot or larger entities such as factories, supply chains, and energy systems [53–55]. In 2021, the International Organization for Standardization (ISO) defined standards for DT framework [56]. The concept of the DT has recently matured into a standardized framework. In this paper's context, DTs enable interaction between the virtual training environment and physical robots.

3.5. Bidirectional Communication

WebSocket protocol provides a single bidirectional Transmission Control Protocol (TCP) connection from a web application to the server [57]. The WebSocket protocol is suitable but not tied to online gaming, stock exchange, and word processing. In the context of the proposed system, WebSocket enables data connections between an MQTT-broker on a cloud server and the user web application. The protocol enables both encrypted and unencrypted connections. In the proposed VEs, an encrypted connection method enables cybersecurity and meets the requirement N3. The WebSocket protocol support was added to the MQTT standard in 2014 [58], enabling MQTT messaging for web applications. In addition to the standard, also OPC Unified Architecture (OPC UA) standard is used to communicate with the hardware. OPC UA is an open cross-platform IEC62541 standard for data exchange [59]. OPC UA is a widely utilized industrial communication standard and Industry 4.0 reference standard [60,61].

3.6. Real-Time Video

Web Real-Time Communication (WebRTC) is an open-source technology for real-time web-based communication [62]. Considering projected real-world elements in the AV experience, delays in video transmission affect the user experience and may cause the user to misguide the robot during a task. Major web browsers support WebRTC on mobile and desktop devices.WebRTC meets the requirement N6 with proven Round Trip Times of 80 to 100 milliseconds on mobile platforms [63]. WebRTC utilizes User Datagram Protocol (UDP) for video data transfer. Since UDP does not support congestion control, the WebRTC server component on the cloud server is required to control the congestion.

3.7. Extended Reality Software

This section evaluates three game engines to choose the game engine to build extended reality training environments meeting the requirement specification defined in Section 2.1. The focus is on ease of distribution and updating of applications, accessibility, and cross-platform support. Epic's Unreal Engine (UE), launched in 1998, is widely utilized in various industrial applications [64]. UE provides methods for programming functionality of VEs:

using C++ or Blueprints. While Blueprint is an easy-to-use visual tool for scripting design, C++ enables programming more complex functionalities. Official releases of UE do not include support to build WebGL runtimes; instead, UE version 4.24 at GitHub supports compiling WebGL runtimes. Unfortunately, the UE GitHub project page was closed during the writing of this paper, preventing utilizing UE4 as a game engine for the proposed VEs.

Unity Technologies released the Unity3D development platform in 2005 as a game engine for macOS. Currently, Unity3D is one of the most popular game engines for desktop and mobile devices [65]. Unity3D includes physics and rendering engines, asset store, and graphical editor [66]. Visual Studio integrated development environment is utilized to program functionality with C#. Unity3D is a cross-platform game engine enabling compiling of XR applications for Windows, macOS, Android, and WebGL. The online asset store provides a wide selection of contributed components as importable assets, enabling functions such as web browsers, interaction toolkits, and connections to various physics engines. For the MQTT connectivity, Best MQTT-asset for Unity3D written by Tivadar György Nagy can be utilized [67].

Open source game engine Godot [68,69] was released under an MIT license in 2014 as an open-source alternative for Unity3D. Godot comes in two versions, the Mono version supports C#, and the native version supports GDScript language. Compiling both C# and GDScript projects is currently supported only for desktop systems; mobile applications support only GDScript. Godot engine provides an asset library and a graphical scene editor. The Godot editor user interface is very similar to the Unity3D editor, providing views of the hierarchy, scene, and properties of objects. Godot supports the cross-platform building of XR applications for Windows, Linux, iOS, macOS, Android, and WebGL [70]. Currently, Godot does not support Websocket MQTT-protocol, and implementing one is beyond the scope of this publication.

Unity3D is the choice for this project since it supports multiple features required to meet the requirement specification. Unity3D enables the compiling of Web-based XR applications, providing platform independence, easy distribution, and accessibility required in N2, N5, N7, and N8. In addition, support for the MQTT protocol enables the bidirectional communication required by N9.

3.8. Extended Reality in Robotic Training

The industry utilizes XR for various industrial training purposes such as mining [71], construction [72,73], and maintenance [74,75]. Applying XR for the training of industrial robot programming and operating tasks has the potential to provide effective training environments. In 1994 Miner et al. [76] developed a VR solution for industrial robot control and operator training for hazardous waste handling. VR solution enabled trainees to create robot trajectories in an intuitive way and, after validation, to upload trajectories to the real robot. Operators tested the solution, and according to their feedback, the system was easy to use. A few years later, Burdea [77] researched the synergy between VR and robotics. Burdea found the potential of VR in the training of robot programming and teleoperation tasks. He described VR robot programming as a user-friendly, high-level programming option for robots compared to the low-level programming method provided by a teach pendant. Furthermore, the robot arm can act as a haptic feedback device, providing force feedback to the user. Crespo et al. [78] implemented and studied a training environment for off-line programming training of an industrial robot. They developed an immersive solution capable of evaluating trajectories created by the user by calculating the shortest collision-free path and comparing the shortest path to the path created by the user. This study supports the results of Miner et al. in terms of enhancing the learning of robot programming and reducing programming time. Perez et al. [79] created a VR application for industrial robot operator training. According to the feedback collected from the pilot users, VR training was user-friendly and reduced the time required for training. The study also proved VR as a risk-free realistic programming method for industrial robots. Monetti et al. [80] compared programming task completion time and task pass rate utilizing VR

and teach pendant on a real robot. The pilot group consisted of twenty-four students who had novice skills in robot programming and performed programming tasks quicker and with a higher pass rate than those utilizing a teach pendant. Dianatfar et al. [81] developed a concept for virtual safety training in human–robot collaboration. The prototype training environment enabled the trainee to visualize the assembly sequence and shared workspace and to predict robot movements. Virtual training enabled comprehensive training of both experienced and inexperienced robot operators. Bogosian et al. [82] demonstrated an AR application enabling construction workers to interact with a virtual robot to learn the basic operation of a robot manipulator. Rofatulhaq et al. [46] developed a web VR application to train engineering students utilizing a virtual robot manipulator. The reviewed papers provide good examples of the ongoing toward utilizing XR for training in the architecture, engineering, and construction sectors (AEC).

4. Implementation and Validation

This section presents the implementation and validation of the VEs. First, we present a description of the system architecture. After describing system architecture, descriptions of individual VE implementations are provided, and finally, validation of VEs. This section provides practical information on implementing VEs similar to the ones presented in this paper.

4.1. System Architecture

The system architecture specification results from requirements specified in Section 2.1 and software components selections explained in Section 3. The central part of the system proposed is a cloud server hosting an MQTT-broker to publish/subscribe messaging between physical and digital twins and to enable congestion control for WebRTC video stream. The Robo3D Lab server connects to the cloud server and acts as a data bridge between hardware and the cloud server. The user device is the device a trainee utilizes to interact with VE and DT. Descriptions of individual VE Implementations are in Sections 4.2.1–4.2.3. Figure 4 describes the connections of the architecture in detail. The user device can be any device capable of running an HTML5-enabled web browser, enabling the trainee to interact with the VE.

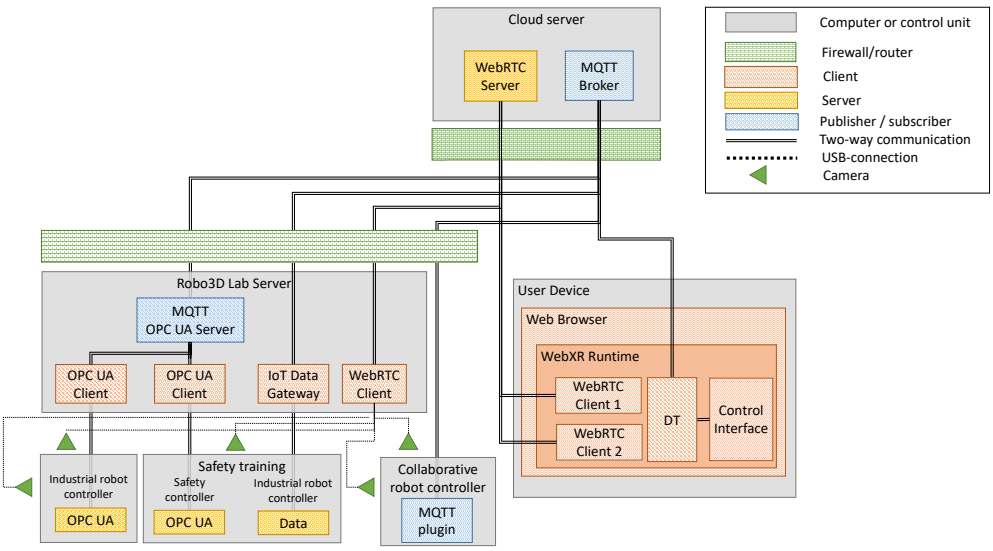

Figure 4. System architecture.

4.2. Robo3D Lab

The implementation of the XR environment started by utilizing an industrial-grade three-dimensional laser scanner to form a point cloud of Robo3D Lab located in Ylivieska, Finland. Based on the point cloud data, a three-dimensional solid model was created and

textured utilizing Blender. Features such as walls, floors, ceilings, pillars, windows, and doors were extruded and cut to match the raw point cloud data. Modeling the basic shapes was followed by overlaying textures to create a realistic virtual training environment. After texturing objects, virtual light sources, reflections, and shadows were added to the scene. Removing the original point cloud data and exporting the model to Unity3D were the last required steps. The three-dimensional model of Robo3D Lab is not a replica; redesigned textures create a modern virtual version.

The Robo3D Lab model was imported into an empty Unity3D project Universal Render Pipeline (URP) template. URP supports cross-platform optimized graphics [83]. The WebXR Exporter [84] asset enables the trainee to interact with the virtual environment, providing required template objects for an in-person player. The VR-toolkit asset enables the teleporting of the player and game objects inside Robo3D Lab. Figure 5 presents a view of the empty Robo3D Lab created with Unity3D as the main VE for the training environment. The following subsections present three robotic training scenarios created inside the main Robo3D Lab VE.

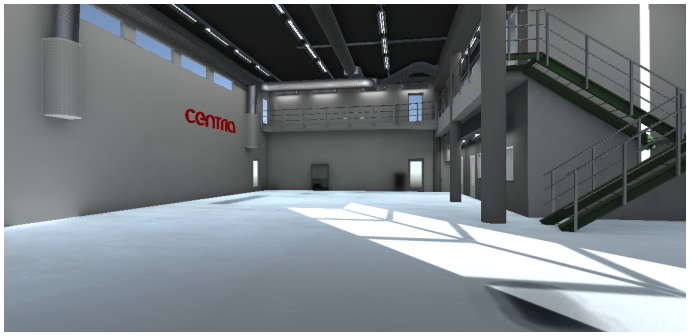

Figure 5. View of Robo3D Lab virtual environment.

4.2.1. Augmented Virtuality Training Scenario

The environment for the AR pick-and-place exercise consists of digital and physical twins of the robot cell located in Centria Robo3D Lab. The physical twin is a collaborative robot (Universal Robots A/S, Odense, Denmark) equipped with an adaptive two-finger gripper. The robot is on top of a table, and the robot control unit is under the tabletop. The objects manipulated in the pick and place training task, a 3D-printed cube, cylinder, and triangular elements, are on the table in front of the robot. Furthermore, a white cardboard box with precut holes for each element is between the robot and the 3D-printed elements.

Three-dimensional model files and dimensional data of the robot arm are required to create a digital model of the robot. Model files and dimensions of the robot arm are from a ROS-industrial universal robot package at GitHub [85]. The teach pendant model file is from the Universal Robots' online library. Model files were modified, textured, and exported to Unity3D utilizing Blender. Furthermore, a screenshot of the Universal Robot jog screen is overlaid on the pendant model to provide an authentic user interface to the trainee. The functionality of the user interface integrates into teach pendant game object, written in C#-script. The functionality is common for arm robots: joint and linear jog buttons inch the robot into the desired direction with a preset velocity. In addition to the jog function, the user interface enables gripper open and close functionality. Blender was utilized to model, texture, and export three-dimensional model files to Unity3D.

As the trainee fully immersed in VR cannot visualize the pose or position of elements, the augmentation of elements from the real world to the virtual experience is required. For this purpose, two web cameras provide a live video stream from top and side perspectives of the 3D-printed elements and cardboard box. The video streams to the XR-training environment utilizing WebRTC over a WebSocket connection. The video projects to the table surface in Unity3D. Projecting positions of the elements and the cardboard box to the trainee. Figure 6 presents a video projected on the table surface. To transfer and project the video onto the texture on the table surface, we utilize code provided by AVStack (AVStack

Pte. Ltd., Singapore) [86]. The accurate position and pose of the elements are required to complete the pick-and-place task successfully.

MQTT-connector provided by 4Each software solutions(4Each s.r.o., Praha, Czech Republic) is installed on the collaborative robot controller to enable bidirectional communication. MQTT-connector enables publishing and subscribing to robot data, enabling communication between physical and digital counterparts.

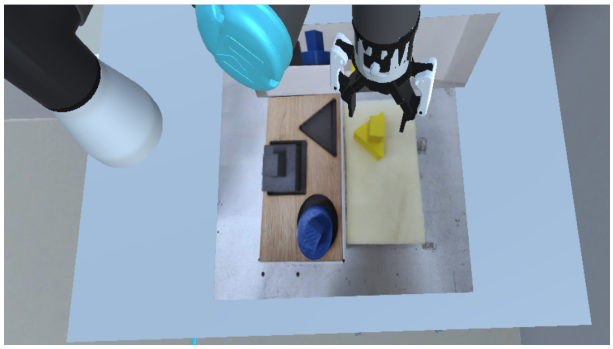

Figure 6. View of Augmented Virtuality setup.

4.2.2. Augmented Reality Training Scenario

The AR implementation consists of digital and physical twins of a Fanuc Educational robot cell (Fanuc Corporation, Oshino-mura, Japan). The robot cell consists of an aluminum profile frame, robot arm, robot controller, machine vision camera, and a two-finger gripper. Additionally, 3D-printed cube elements are included as objects to manipulate. Model files and dimensions of the robot arm are from the ROS-industrial universal robot package at GitHub [85]. Model files of robot 3D-printed elements for exercise were created, textured, and exported to Unity3D utilizing Blender.

The user interface to control the robot utilizes a touch screen typical for mobile devices. The trainee can drag the robot into different positions by simply dragging Tool Center Point (TCP) to the desired location. Buttons overlaid on the AR scene enable the gripper open and close functionality. The functionality of the touch-based user interface to control the robot and the gripper is written in C#. The touch interface provides an intuitive way to control the projected AR robot. In addition to the augmented virtual robot, the trainee can observe a live video stream of the physical robot.

Fanuc Robot controller does not support the MQTT protocol; instead, the robot controller supports the industry-standard OPC UA protocol. A custom OPC UA to MQTT-bridge based on Open62541 [87] enables bidirectional communication between the digital and physical twin of the robot. Figure 7 presents the implemented AR visualization of the robot arm.

Figure 7. Augmented Reality with visualization of robot working area.

4.2.3. Virtual Reality Safety Training Scenario

The physical twin is an industrial production cell consisting of an ABB IRB6700 industrial robot (ABB Ltd., Zurich, Switzerland), safety fences, a safety control panel, safety light curtains, safety laser scanners, and an apartment floor structure. Model files for the robot were obtained from the manufacturer's website and imported into Unity3D with Blender. Model files for the safety light curtains, laser scanners, and safety control panel were created and textured utilizing Blender. A local building manufacturer provided a digital model of the apartment floor structure in an industry-standard Building Information Model (BIM) format. Blender was utilized to import the BIM texture elements of the floor structure and export the textured floor structure into Unity3D. The safety control panel three-dimensional model exported to Unity3D Scene is overlaid with a screenshot of the physical safety control panel. The overlaid screenshot provides the user interface for controlling the robot and safety device functions. The functionality of the safety controller consists of resetting the tripped safety devices and restarting the robot's operation cycle. Violation of the robot operation area is possible from three directions, two safety light curtains and a laser scanner monitor mentioned directions. Figure 8 presents the setup of safety training VE.

ABB Robot controller option "IoT Data Gateway" enables OPC UA and MQTT protocols. The ABB "IoT Data Gateway" option requires client software to act as a gateway between the robot controller and the internet. The ABB client software is installed on the Robo3D Lab server and configured to publish and subscribe robot positions, speed, and gripper states to and from the MQTT broker installed on the cloud server.

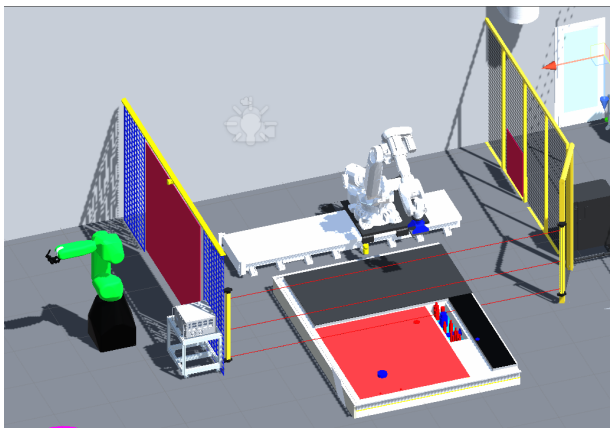

Figure 8. View of Unity 3D scene for safety training.

4.2.4. Digital Twinning

DTs provide the trainee with a way to interact with hardware connected to VEs. Each of the aforementioned digital robots has a physical counterpart and connection in between. DTs of the industrial robots are based on Unity3D Inverse Kinematics (IK). The inverse kinematics solution utilizes the Jacobian matrix to calculate joint values to achieve the requested TCP position. The forward kinematic solution utilizes Denavit–Hartenberg (D-H) convention. Publications presenting utilization of both forward and inverse kinematics calculations for various robot arms are available [88–91] and are used as a basis for the solution presented here. Figure 9 presents the inverse kinematics logic utilized in this work. The rigging of the industrial robot mechanics was created with Blender and imported into Unity3D along with model files. The rigging was utilized in Unity3D to create a kinematic model to bind robot arm mechanical elements together with constraints equivalent to the physical robot. IK calculations are based on the model, preventing moving the robot to non-wanted or out-of-reach positions. The table and walls surrounding the physical robot were also created in Unity3D and defined as physical barriers, limiting the robot's movement inside these barrier elements. DTs presented enable three levels of integration

with physical twins: digital model, shadow, and twin. The trainee can select between the three levels of integration for the DTs, utilizing the XR interface.

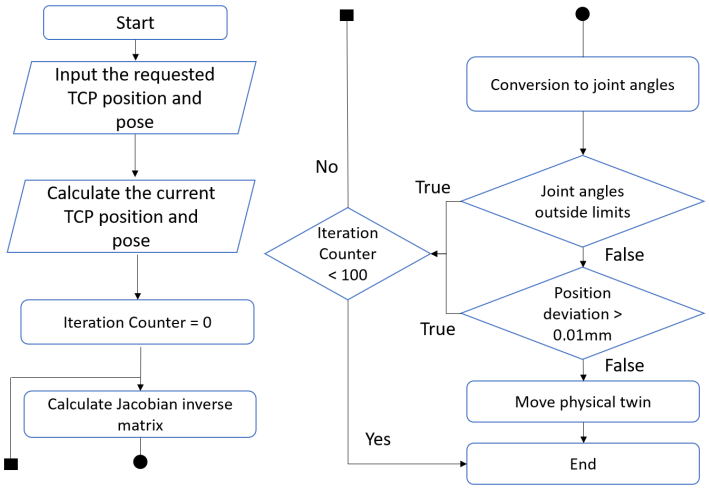

Figure 9. Flowchart of inverse kinematics logic.

Digital and physical twins of industrial robots communicate utilizing the MQTT protocol. Communication utilizes Best MQTT-asset and C#-language. Example code files provided with assets were modified to create a bidirectional communication link between the twins. MQTT is the main protocol of this implementation; however, the implementation does not limit to MQTT. The implementation supports OPC UA and proprietary communication protocols by utilizing a custom bridge described in Section 4.2.2. Enabling the connection to virtually any production machine with control over Ethernet communication.

4.3. Functionality Validation

To validate the functionality of the XR training environment, a high-level task was defined for each of the aforementioned scenarios. Validations were performed against the defined requirement specification defined in Section 2.1. To validate cross-platform support of the application, HTC Vice and Oculus Quest devices were utilized for immersive VEs. iOS tablet device and Android phone were utilized in the functionality validation of the AR implementation.

4.3.1. Augmented Virtuality Training Scenario

For the collaborative robot, we defined a simple pick-and-place task. The goal for the validation task is to pick up the cube, cylinder, and triangular objects from the table and place them into the precut holes of the cardboard box. The task requires programming the robot to approach, pick, place, retract, and control the gripper functions.

The initial functionality validation failed because controller button definitions differ between Oculus and Vive: the physical button layout is not identical. Button definitions in Unity3D C#-scripts were re-implemented to support controllers from both manufacturers. All the functions, such as teleporting and jogging of the robot, are mapped to the joystick on Oculus controllers and touchpad on Vive controllers, providing a similar user experience for both devices. The first implementation for jogging the robot enabled controlling the robot by inching the DT. Due to delays in data transfer, this implementation method resulted in a poor user experience. Replacing the inching method with a method enabling the trainee to initiate the DT movement after inching the digital model of the robot to the desired end position resolved the issue. If the DT validates the end position initiated by the trainee, the end position passes to the physical robot. The mentioned implementation was accepted and chosen for the final evaluation. Validation pointed out that the gray color of the cube element was hard to distinguish from the black-colored gripper. Replacing the gray cube element with a red cube simplifies the training task. After re-implementing

controller button mappings, the color of the cube, and the method for the jogging of the robot, validation of the VE was successful.

4.3.2. Augmented Reality Training Scenario

A three-dimensional AR model of the robot, gripper, 3D-printed elements, and wire-frame visualization of the robot's working area form a training task for the industrial robot. This task aims to assemble the 3D-printed cube, triangle, and cylinder objects into a pile utilizing tabs and notches printed onto the objects. The trainee can augment the AR model to any physical environment and create a program for the robot to manipulate the cube elements. This task requires program the robot to approach, pick, place, retract, and control the gripper functions. The challenging part of this exercise is the short reach of the robot, requiring accurate planning of robot trajectories to pile all 3D-printed elements on top of each other.

The initial validation for this exercise failed because controlling the robot with the touch screen is inaccurate, making it impossible to accurately position the 3D-printed elements on top of each other. This part of the exercise was re-implemented by creating snap points for each of the 3D-printed elements in Unity3D. Snap points enabled the trainee to guide the robot by clicking pre-defined points in the scene. After adding snap points, the validation was successful.

4.3.3. Virtual Reality Safety Training Scenario

A human–robot collaboration training task was defined to validate the functionality of the safety training VE. The task requires the trainee to insert HVAC piping and a layer of glue onto the floor structure during the apartment floor manufacturing process. The trainee is signaled by a green light when to enter the robot cell to add glue and HVAC-piping between the robot work cycles. After performing manual operations, the trainee must press a button on the safety controller screen to continue the robot's operation. If the trainee interferes with the robot cell area during the robot work cycle, the safety device is triggered, and the robot motion stops. The trainee must reset the alarms on the safety controller and the robot controller to restart the robot.

The initial validation failed because the triggering of virtual safety devices caused the physical robot to halt during motion. The implementation was modified so that the physical robot does not halt; instead of causing a halt on the physical robot, only the DT halts. After resetting the virtual safety controller and restarting, the DT continues synchronizing its motion with the physical robot. The validation was successful after this modification.

5. Discussion

This paper presented three XR robotic training scenarios designed to support the digitally oriented learning process of digital-native trainees. To enable location and time independence, web-based VEs were designed and implemented. MQTT protocol enables bidirectional communication between the digital and physical twins of the robots. Figure 10 presents the configuration of the physical and digital training environment. To validate the functionality of developed VEs, prototypes of training environments were created and tested. Prototypes of VEs presented open up a discussion on the benefits and challenges of utilizing DTs and XR for robotics training. The key findings of the authors are listed in the following subsections.

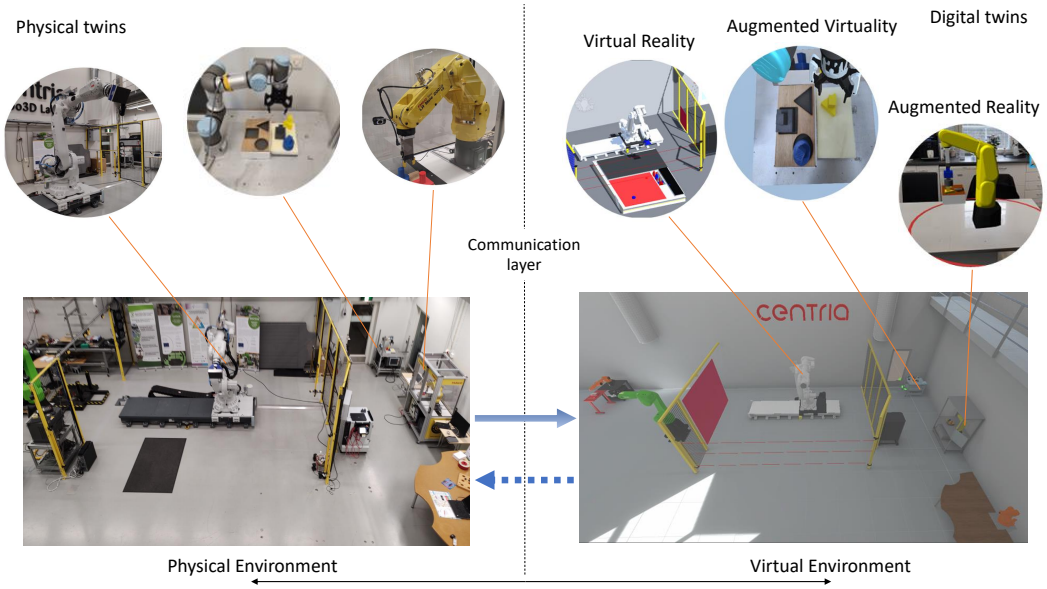

Figure 10. Physical and virtual training environment.

5.1. Augmented Virtuality Enables Virtual Teleoperation

AV enables the augmenting of real-world elements to the virtual world. The added value is providing interactive object data from the real world to the trainee fully immersed in VR. In addition, AV enables environment-aware teleoperation of industrial equipment such as robot manipulators. In this publication, AV implementation augmented side and top views of real-world elements to VR. The proposed method enables near real-time update of element positions and effective user experience. This exercise allows the trainee to fully immerse into a high-fidelity environment provided by VR and concentrate on the exercise itself rather than spending time and effort understanding the platforms. Furthermore, DT enables interaction which utilizes real hardware to perform specific tasks, such as picking and placing 3D-printed objects with an industrial robot manipulator.

5.2. Augmented Reality Enables Mobility

AR enables the visualization of virtual elements regardless of location. Modern mobile devices such as phones and tablets provide support to experience WebXR applications. Therefore, AR-based training enables the trainee to visualize the robot workspace and safety areas in any environment at any time. Besides that, the touch user interface (TUI) provides an easy-to-learn element for interacting with the robot. For this reason, the added value of AR for remote robotics training is the independence of device, time, and location. DTs provide a way to visualize and validate designed trajectories utilizing real hardware, enabling almost hands-on experience for the trainee at a reduced cost and simplified logistics.

5.3. Virtual Reality Enables Risk-Free Safety Training

VR provides an immersive and risk-free method for safety training. Safety perimeters and sensing elements of safety devices, such as laser beams not visible in the real world, can be visualized in VE. Exposing invisible safety features to the trainee enables a clear understanding of areas to be avoided during the robot operation. The added value in VR safety training is the possibility to visually inspect elements invisible in reality and a risk-free way of exercising collaboration with industrial robots. In addition, VR reduces restrictions related to safety regulations by not exposing trainees to hazardous situations commonly found in industrial environments. The added value of the DT is that the training utilizes real-world data from the physical twin, providing a more realistic virtual experience to the trainee compared to a simulated environment.

5.4. Towards Metaverse and Aquatic Environment

DTs and VEs proposed in this paper can be utilized as core elements for Metaverse training environments in the future. The multi-user nature of the Metaverse enables social interaction between trainees and trainers of the VEs, bringing the training experience closer to training in physical environments. Furthermore, the Metaverse converges digital-natives natural with virtual environments with digital industrial training platforms. We intend also to add a virtual aquatic environment based on the northern part of Biscayne Bay, Florida, USA, to demonstrate the generalization of the proposed virtual framework. The range of robot types will expand to include underwater robots and unmanned surface vehicles. One of the primary purposes of such a virtual environment is to provide tools for testing and validating underwater robot navigation algorithms and persistent monitoring of aquatic ecosystems.

6. Conclusions

Digital twin-based VEs for realistic robotic training were proposed in this paper. To the best of our knowledge, the full capacity of DTs with a true bidirectional information transfer has not been fully utilized in robotics training before. Extensively implemented DTs, integrated into training, enable trainees to interact with real hardware and experience the near hands-on sensation of controlling a robot. In conclusion, the added value of XR in robotics training is multifaceted.

AV enables the immersive teleoperation of robotic systems by augmenting interactable physical objects into the virtual world. In addition, AR supports mobility, enabling robotics training anywhere, anytime, with any device. Furthermore, VR introduces immersive, risk-free safety training for robotic applications, relaxing constraints related to safety regulations. In conclusion, cyberspaces not bound by limitations of reality, such as the physical locations of the trainee, trainer, or robot, provide a natural way for the digital-native generation to embrace robotics.

In future work, we intend to add a multi-user capability to enable natural collaboration and social interaction for the digital-native generation. An accessible multi-user training environment enables a trivial way to experience the Metaverse, in aquatic and nonaquatic robotic training.

Author Contributions: conceptualization, T.K., S.P., L.B., P.P. and T.P.; methodology, T.K. and T.P.; software, T.K.; validation, T.K. and P.P.; investigation, T.K., S.P. and L.B.; data curation, T.K. and T.P.; writing—original draft preparation, T.K., T.P., L.B. and P.P.; writing—review and editing, T.K., T.P. and P.P.; visualization, T.K.; supervision, S.P. and L.B.; project administration, T.K. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation program (grant agreement n° 825196), the NSF grants IIS-2034123, IIS-2024733, and by the U.S. Dept. of Homeland Security 2017-ST-062000002.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Miranda, J.; Navarrete, C.; Noguez, J.; Molina-Espinosa, J.M.; Ramírez-Montoya, M.S.; Navarro-Tuch, S.A.; Bustamante-Bello, M.R.; Rosas-Fernández, J.B.; Molina, A. The core components of education 4.0 in higher education: Three case studies in engineering education. *Comput. I Electr. Eng.* **2021**, *93*, 107278. [CrossRef]
2. Prensky, M. Digital Natives, Digital Immigrants. 2001. Available online: https://www.marcprensky.com/writing/Prensky%20-%20Digital%20Natives,%20Digital%20Immigrants%20-%20Part1.pdf (accessed on 2 October 2022).
3. Downes, J.; Bishop, P. Educators Engage Digital Natives and Learn from Their Experiences with Technology: Integrating Technology Engages Students in Their Learning. *Middle Sch. J.* **2012**, *43*, 6–15. [CrossRef]

4. Cheng, W.; Chen, P.; Liu, X.; Huang, R. Designing Authentic Learning to Meet the Challenges of Digital Natives in First-Year Program: An Action Research in Chinese University. In Proceedings of the 2016 IEEE 16th International Conference on Advanced Learning Technologies (ICALT), Austin, TX, USA, 25–28 July 2016; pp. 453–454.
5. Rideout, V.; Foehr, U.; Roberts, D. GENERATION M2 Media in the Lives of 8- to 18-Year-Olds. 2010. Available online: https://files.eric.ed.gov/fulltext/ED527859.pdf (accessed on 2 October 2022).
6. Lee, H.J.; Gu, H.H. Empirical Research on the Metaverse User Experience of Digital Natives. *Sustainability* **2022**, *14*, 4747. [CrossRef]
7. Alnagrat, A.J.A.; Ismail, R.C.; Idrus, S.Z.S. Extended Reality (XR) in Virtual Laboratories: A Review of Challenges and Future Training Directions. *J. Phys. Conf. Ser.* **2021**, *1874*, 012031. [CrossRef]
8. Harris, A.; Rea, A. Web 2.0 and Virtual World Technologies: A Growing Impact on IS Education. *J. Inf. Syst. Educ.* **2009**, *20*, 137–144. [CrossRef]
9. Milgram, P.; Kishino, F. A Taxonomy of Mixed Reality Visual Displays. *IEICE Trans. Inf. Syst.* **1994**, *E77-D*, 1321–1329.
10. Heilig, M.L. Sensorama Simulator. U.S. Patent 3050870A, 10 January 1961.
11. Feisel, L.D.; Rosa, A.J. The role of the laboratory in undergraduate engineering education. *J. Eng. Educ.* **2005**, *94*, 121–130. [CrossRef]
12. Jeršov, S.; Tepljakov, A. Digital twins in extended reality for control system applications. In Proceedings of the 2020 IEEE 43rd International Conference on Telecommunications and Signal Processing (TSP), Milan, Italy, 7–9 July 2020; pp. 274–279.
13. Qi, Q.; Tao, F. Digital Twin and Big Data Towards Smart Manufacturing and Industry 4.0: 360 Degree Comparison. *IEEE Access* **2018**, *6*, 3585–3593. [CrossRef]
14. Li, X.; He, B.; Zhou, Y.; Li, G. Multisource Model-Driven Digital Twin System of Robotic Assembly. *IEEE Syst. J.* **2021**, *15*, 114–123. [CrossRef]
15. Li, X.; He, B.; Wang, Z.; Zhou, Y.; Li, G.; Jiang, R. Semantic-Enhanced Digital Twin System for Robot–Environment Interaction Monitoring. *IEEE Trans. Instrum. Meas.* **2021**, *70*, 1–13. [CrossRef]
16. Garg, G.; Kuts, V.; Anbarjafari, G. Digital Twin for FANUC Robots: Industrial Robot Programming and Simulation Using Virtual Reality. *Sustainability* **2021**, *13*, 10336. [CrossRef]
17. Kim, Y.; Lee, S.y.; Lim, S. Implementation of PLC controller connected Gazebo-ROS to support IEC 61131-3. In Proceedings of the 2020 25th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), Vienna, Austria, 8–11 September 2020; Volume 1, pp. 1195–1198.
18. Shen, B.; Xia, F.; Li, C.; Martín-Martín, R.; Fan, L.; Wang, G.; Pérez-D'Arpino, C.; Buch, S.; Srivastava, S.; Tchapmi, L.; et al. iGibson 1.0: A Simulation Environment for Interactive Tasks in Large Realistic Scenes. In Proceedings of the 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Prague, Czech Republic, 27 September–1 October 2021; pp. 7520–7527.
19. What Is the Industrial Metaverse—And Why Should I Care? 2022. Available online: https://new.siemens.com/global/en/company/insights/what-is-the-industrial-metaverse-and-why-should-i-care.html (accessed on 4 November 2022).
20. Mystakidis, S. Metaverse. *Encyclopedia* **2022**, *2*, 486–497. [CrossRef]
21. Six Trailblazing Use Cases for the Metaverse in Business. 2022. Available online: https://www.nokia.com/networks/insights/metaverse/six-metaverse-use-cases-for-businesses/ (accessed on 4 November 2022).
22. Kaarlela, T.; Arnarson, H.; Pitkäaho, T.; Shu, B.; Solvang, B.; Pieskä, S. Common Educational Teleoperation Platform for Robotics Utilizing Digital Twins. *Machines* **2022**, *10*, 577. [CrossRef]
23. Heilig, M.L. EL Cine del Futuro: The Cinema of the Future. *Presence Teleoperators Virtual Environ.* **1992**, *1*, 279–294.
24. Coiffet, P.; Burdea, G. *Virtual Reality Technology*; IEEE Press: Hoboken, NJ, USA; Wiley: Hoboken, NJ, USA, 2017.
25. LaValle, S.M. *Virtual Reality*; Cambridge University Press: Cambridge, UK, 2020.
26. Robinett, W.; Rolland, J.P. 5—A Computational Model for the Stereoscopic Optics of a Head-Mounted Display. In *Virtual Reality Systems*; Earnshaw, R., Gigante, M., Jones, H., Eds.; Academic Press: Boston, MA, USA, 1993; pp. 51–75. [CrossRef]
27. Regrebsubla, N. *Determinants of Diffusion of Virtual Reality*; GRIN Verlag: Munich, Germany, 2016.
28. Steinicke, F. *Being Really Virtual*; Springer: Berlin/Heidelberg, Germany, 2016.
29. Sutherland, I.E. A Head-Mounted Three Dimensional Display. In Proceedings of the Fall Joint Computer Conference, Part I—Association for Computing Machinery, AFIPS '68 (Fall, Part I), New York, NY, USA, 9–11 December 1968; pp. 757–764. [CrossRef]
30. MecoLLuM, H.J.D.N. Stereoscopic Television Apparatus. U.S. Patent 2388170A, 30 October 1945.
31. Heilig, M.L. Stereoscopic-Television Apparatus for Individual Use. U.S. Patent 2955156A, 24 May 1957.
32. Bradley, W.E. Remotely Controlled Remote Viewing System. U.S. Patent 3205303A, 27 March 1961.
33. Caudell, T.; Mizell, D. Augmented reality: An application of heads-up display technology to manual manufacturing processes. In Proceedings of the Twenty-Fifth Hawaii International Conference on System Sciences, Kauai, HI, USA, 7–10 January 1992; Volume 2, pp. 659–669.
34. Krueger, M.W. Responsive Environments. In Proceedings of the National Computer Conference—Association for Computing Machinery, AFIPS '77, New York, NY, USA, 13–16 June 1977; pp. 423–433. [CrossRef]
35. Howlett, E.M. Wide Angle Color Photography Method and System. U.S. Patent 4406532A, 27 September 1983.
36. Kawamoto, H. The history of liquid-crystal display and its industry. In Proceedings of the 2012 Third IEEE HISTory of ELectro-Technology CONference (HISTELCON), Pavia, Italy, 5–7 September 2012; pp. 1–6. [CrossRef]

37. Kalawsky, R.S. Realities of using visually coupled systems for training applications. In Proceedings of the Helmet-Mounted Displays III, Orlando, FL, USA, 21–22 April 1992; Volume 1695.
38. LaValle, S.M.; Yershova, A.; Katsev, M.; Antonov, M. Head tracking for the Oculus Rift. In Proceedings of the 2014 IEEE International Conference on Robotics and Automation (ICRA), Hong Kong, China, 31 May–7 June 2014; pp. 187–194. [CrossRef]
39. Borrego, A.; Latorre, J.; Alca niz, M.; Llorens, R. Comparison of Oculus Rift and HTC Vive: Feasibility for Virtual Reality-Based Exploration, Navigation, Exergaming, and Rehabilitation. *Games Health J.* **2018**, *7*, 151–156. [CrossRef] [PubMed]
40. Oculus Device Specification. 2022. Available online: https://developer.oculus.com/resources/oculus-device-specs/ (accessed on 2 September 2022).
41. Introducing Varjo XR-3. 2022. Available online: https://varjo.com/products/xr-3/ (accessed on 2 September 2022).
42. Buy Vive Hardware. 2022. Available online: https://www.vive.com/eu/product/vive/ (accessed on 31 August 2022).
43. Hickson, I.; Hyatt, D. HTML 5. 2008. Available online: https://www.w3.org/TR/2008/WD-html5-20080122/ (accessed on 30 October 2022).
44. Khan, M.Z.; Hashem, M.M.A. A Comparison between HTML5 and OpenGL in Rendering Fractal. In Proceedings of the 2019 International Conference on Electrical, Computer and Communication Engineering (ECCE), Chittagong, Bangladesh, 7–9 February 2019; pp. 1–6. [CrossRef]
45. Maclntyre, B.; Smith, T.F. Thoughts on the Future of WebXR and the Immersive Web. In Proceedings of the 2018 IEEE International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct), Munich, Germany, 16–20 October 2018; pp. 338–342. [CrossRef]
46. Rofatulhaq, H.; Wicaksono, S.A.; Falah, M.F.; Sukaridhoto, S.; Zainuddin, M.A.; Rante, H.; Al Rasyid, M.U.H.; Wicaksono, H. Development of Virtual Engineering Platform for Online Learning System. In Proceedings of the 2020 International Conference on Computer Engineering, Network, and Intelligent Multimedia (CENIM), Surabaya, Indonesia, 17–18 November 2020; pp. 185–192. [CrossRef]
47. WebAssembly Is Here! 2018. Available online: https://blog.unity.com/technology/webassembly-is-here (accessed on 20 September 2022).
48. Gonzâles, J. mod_wasm: Run WebAssembly with Apache. 2022. Available online: https://wasmlabs.dev/articles/apache-mod-wasm/ (accessed on 30 October 2022).
49. Grieves, M. *Origins of the Digital Twin Concept*; Florida Institute of Technology: Melbourne, FL, USA, 2016; Volume 8. [CrossRef]
50. Cimino, C.; Negri, E.; Fumagalli, L. Review of digital twin applications in manufacturing. *Comput. Ind.* **2019**, *113*, 103130. [CrossRef]
51. Kritzinger, W.; Karner, M.; Traar, G.; Henjes, J.; Sihn, W. Digital Twin in manufacturing: A categorical literature review and classification. *IFAC-PapersOnLine* **2018**, *51*, 1016–1022. [CrossRef]
52. Digital Twin Consortium. Glossary of Digital Twins. 2021. Available online: https://www.digitaltwinconsortium.org/glossary/glossary.html#digital-twin (accessed on 26 November 2022).
53. Gomez, F. AI-Driven Digital Twins and the Future of Smart Manufacturing. 2021. Available online: https://www.machinedesign.com/automation-iiot/article/21170513/aidriven-digital-twins-and-the-future-of-smart-manufacturing (accessed on 26 November 2022).
54. How Digital Twins Are Driving the Future of Engineering. 2021. Available online: https://www.nokia.com/networks/insights/technology/how-digital-twins-driving-future-of-engineering/ (accessed on 20 November 2022).
55. Fang, L.; Liu, Q.; Zhang, D. A Digital Twin-Oriented Lightweight Approach for 3D Assemblies. *Machines* **2021**, *9*, 231. [CrossRef]
56. IOS. *Automation Systems and Integration—Digital Twin Framework for Manufacturing—Part 1: Overview and General Principles*; International Organization for Standardization: Geneva, Switzerland, 2000.
57. The WebSocket Protocol. 2011. Available online: https://www.rfc-editor.org/rfc/rfc6455.html (accessed on 6 September 2022).
58. MQTT Version 3.1.1. 2014. Available online: https://docs.oasis-open.org/mqtt/mqtt/v3.1.1/os/mqtt-v3.1.1-os.pdf (accessed on 14 September 2022).
59. OPCFoundation. What Is OPC? 2022. Available online: https://opcfoundation.org/about/what-is-opc/ (accessed on 3 November 2022).
60. Mühlbauer, N.; Kirdan, E.; Pahl, M.O.; Carle, G. Open-Source OPC UA Security and Scalability. In Proceedings of the 2020 25th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), Vienna, Austria, 8–11 September 2020; Volume 1, pp. 262–269.
61. Drahoš, P.; Kučera, E.; Haffner, O.; Klimo, I. Trends in industrial communication and OPC UA. In Proceedings of the 2018 Cybernetics Informatics (K I), Lazy pod Makytou, Slovakia, 31 January–3 February 2018; pp. 1–5. 8337560. [CrossRef]
62. Eltenahy, S.; Fayez, N.; Obayya, M.; Khalifa, F. Comparative Analysis of Resources Utilization in Some Open-Source Videoconferencing Applications based on WebRTC. In Proceedings of the 2021 International Telecommunications Conference (ITC-Egypt), Alexandria, Egypt, 13–15 July 2021; pp. 1–4. [CrossRef]
63. Jansen, B.; Goodwin, T.; Gupta, V.; Kuipers, F.; Zussman, G. Performance Evaluation of WebRTC-based Video Conferencing. *ACM SIGMETRICS Perform. Eval. Rev.* **2018**, *45*, 56–68. [CrossRef]

64. Peters, E.; Heijligers, B.; de Kievith, J.; Razafindrakoto, X.; van Oosterhout, R.; Santos, C.; Mayer, I.; Louwerse, M. Design for Collaboration in Mixed Reality: Technical Challenges and Solutions. In Proceedings of the 2016 8th International Conference on Games and Virtual Worlds for Serious Applications (VS-GAMES), Barcelona, Spain, 7–9 September 2016; pp. 1–7.
65. Datta, S. Top Game Engines To Learn in 2022. 2022. Available online: https://blog.cloudthat.com/top-game-engines-learn-in-2022/ (accessed on 5 November 2022).
66. Juliani, A.; Berges, V.P.; Teng, E.; Cohen, A.; Harper, J.; Elion, C.; Goy, C.; Gao, Y.; Henry, H.; Mattar, M.; et al. Unity: A general platform for intelligent agents. *arXiv* **2018**, arXiv:1809.02627.
67. Best MQTT. 2022. Available online: https://assetstore.unity.com/packages/tools/network/best-mqtt-209238 (accessed on 23 November 2022).
68. Linietsky, J.; Manzur, A. Godot Engine. 2022. Available online: https://github.com/godotengine/godot (accessed on 5 November 2022).
69. Thorn, A. *Moving from Unity to Godot*; Springer: Berlin/Heidelberg, Germany, 2020.
70. Linietsky, J.; Manzur, A. GDScript Basics. 2022. Available online: https://docs.godotengine.org/en/stable/tutorials/scripting/gdscript/gdscript_basics.html (accessed on 5 November 2022).
71. Pomykała, R.; Cybulski, A.; Klatka, T.; Patyk, M.; Bonieckal, J.; Kedzierski, M.; Sikora, M.; Juszczak, J.; Igras-Cybulska, M. "Put your feet in open pit"—A WebXR Unity application for learning about the technological processes in the open pit mine. In Proceedings of the 2022 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW), Christchurch, New Zealand, 12–16 March 2022; pp. 493–496.
72. Rokooei, S.; Shojaei, A.; Alvanchi, A.; Azad, R.; Didehvar, N. Virtual reality application for construction safety training. *Saf. Sci.* **2023**, *157*, 105925. [CrossRef]
73. Moore, H.F.; Gheisari, M. A Review of Virtual and Mixed Reality Applications in Construction Safety Literature. *Safety* **2019**, *5*, 51. [CrossRef]
74. Kunnen, S.; Adamenko, D.; Pluhnau, R.; Loibl, A.; Nagarajah, A. System-based concept for a mixed reality supported maintenance phase of an industrial plant. *Procedia CIRP* **2020**, *91*, 15–20. . [CrossRef]
75. Ariansyah, D.; Erkoyuncu, J.A.; Eimontaite, I.; Johnson, T.; Oostveen, A.M.; Fletcher, S.; Sharples, S. A head mounted augmented reality design practice for maintenance assembly: Toward meeting perceptual and cognitive needs of AR users. *Appl. Ergon.* **2022**, *98*, 103597. [CrossRef] [PubMed]
76. Miner, N.; Stansfield, S. An interactive virtual reality simulation system for robot control and operator training. In Proceedings of the 1994 IEEE International Conference on Robotics and Automation, San Diego, CA, USA, 8–13 May 1994; Volume 2, pp. 1428–1435.
77. Burdea, G. Invited review: The synergy between virtual reality and robotics. *IEEE Trans. Robot. Autom.* **1999**, *15*, 400–410. [CrossRef]
78. Crespo, R.; Garcia, R.; Quiroz, S. Virtual Reality Application for Simulation and Off-line Programming of the Mitsubishi Movemaster RV-M1 Robot Integrated with the Oculus Rift to Improve Students Training. *Procedia Comput. Sci.* **2015**, *75*, 107–112.
79. Pérez, L.; Diez, E.; Usamentiaga, R.; García, D.F. Industrial robot control and operator training using virtual reality interfaces. *Comput. Ind.* **2019**, *109*, 114–120. [CrossRef]
80. Monetti, F.; de Giorgio, A.; Yu, H.; Maffei, A.; Romero, M. An experimental study of the impact of virtual reality training on manufacturing operators on industrial robotic tasks. *Procedia CIRP* **2022**, *106*, 33–38.
81. Dianatfar, M.; Latokartano, J.; Lanz, M. Concept for Virtual Safety Training System for Human-Robot Collaboration. *Procedia Manuf.* **2020**, *51*, 54–60. [CrossRef]
82. Bogosian, B.; Bobadilla, L.; Alonso, M.; Elias, A.; Perez, G.; Alhaffar, H.; Vassigh, S. Work in Progress: Towards an Immersive Robotics Training for the Future of Architecture, Engineering, and Construction Workforce. In Proceedings of the 2020 IEEE World Conference on Engineering Education (EDUNINE), Bogota, Colombia, 15–18 March 2020; pp. 1–4. [CrossRef]
83. Render Pipelines. 2022. Available online: https://docs.unity3d.com/2019.3/Documentation/Manual/render-pipelines.html (accessed on 20 November 2022).
84. Weizman, O. WebXR Export. 2018. Available online: https://github.com/De-Panther/unity-webxr-export (accessed on 26 September 2022).
85. Messmer, F.; Hawkins, K.; Edwards, S.; Glaser, S.; Meeussen, W. Universal Robot. 2019. Available online: https://github.com/ros-industrial/universal_robot (accessed on 26 September 2022).
86. Hugo, J. Jitsi-Meet-Unity-Demo. 2021. Available online: https://github.com/avstack/jitsi-meet-unity-demo (accessed on 11 December 2022).
87. Pfrommer, J. open62541. 2019. Available online: https://github.com/open62541/open62541 (accessed on 3 November 2022).
88. Barakat, A.N.; Gouda, K.A.; Bozed, K.A. Kinematics analysis and simulation of a robotic arm using MATLAB. In Proceedings of the 2016 4th International Conference on Control Engineering & Information Technology (CEIT), Hammamet, Tunisia, 16–18 December 2016; pp. 1–5.
89. Michas, S.; Matsas, E.; Vosniakos, G.C. Interactive programming of industrial robots for edge tracing using a virtual reality gaming environment. *Int. J. Mechatron. Manuf. Syst.* **2017**, *10*, 237. [CrossRef]

90. Nicolescu, A.; Ilie, F.M.; Tudor George, A. Forward and inverse kinematics study of industrial robots taking into account constructive and functional parameter's modeling. *Proc. Manuf. Syst.* **2015**, *10*, 157.
91. Craig, J.J. *Introduction to Robotics: Mechanics and Control*; Pearson Educacion: London, UK, 2005.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.