applied sciences

MDPI

Article

Integrating Virtual, Mixed, and Augmented Reality to Human–Robot Interaction Applications Using Game Engines: A Brief Review of Accessible Software Tools and Frameworks

Enrique Coronado *, Shunki Itadera and Ixchel G. Ramirez-Alpizar

National Institute of Advanced Industrial Science and Technology (AIST), Tokyo 135-0064, Japan
* Correspondence: coronadozuniga.enrique@aist.go.jp

Abstract: This article identifies and summarizes software tools and frameworks proposed in the Human–Robot Interaction (HRI) literature for developing extended reality (XR) experiences using game engines. This review includes primary studies proposing Virtual Reality (VR), Augmented Reality (AR), and Mixed Reality (MR) solutions where humans can control or interact with real robotic platforms using devices that extend the user's reality. The objective of this article is not to present an extensive list of applications and tools. Instead, we present recent, relevant, common, and accessible frameworks and software tools implemented in research articles published in high-impact robotics conferences and journals. For this, we searched papers published during a seven-years period between 2015 and 2022 in relevant databases for robotics (Science Direct, IEEE Xplore, ACM digital library, Springer Link, and Web of Science). Additionally, we present and classify the application context of the reviewed articles in four groups: social robotics, programming of industrial robots, teleoperation of industrial robots, and Human–Robot collaboration (HRC).

Keywords: Human–Robot Interaction; robotics; game engine; Virtual Reality; Mixed Reality; Augmented Reality; extended reality; Cyber-Physical Systems

Citation: Coronado, E.; Itadera, S.; Ramirez-Alpizar, I.G. Integrating Virtual, Mixed, and Augmented Reality to Human–Robot Interaction Applications Using Game Engines: A Brief Review of Accessible Software Tools and Frameworks. *Appl. Sci.* **2023**, *13*, 1292. https://doi.org/10.3390/app13031292

Academic Editor: Vassilis Charissis

Received: 21 December 2022
Revised: 10 January 2023
Accepted: 12 January 2023
Published: 18 January 2023

1. Introduction

Extended Reality (XR) is an umbrella concept encompassing the entire spectrum of methods able to alter reality through immersive technology, such as Virtual Reality (VR), Augmented Reality (AR), and Mixed Reality (MR) [1]. In robotics, XR applications can be developed on top of robot-specific simulators or game engines [2]. Robot-specific simulators are generally used for offline testing of robotic systems. As presented in [3,4], certain robot-specific simulators enable the creation of virtual reality environments. However, robot-specific simulators can limit the content, graphics quality, and supported hardware of developed applications. To explore the capabilities of XR technologies, researchers require software platforms compatible with new XR hardware that facilitates the generation of content for these devices as well as other programming and integration tasks [2]. In this context, game engines such as Unity and Unreal Engine provide several tools to facilitate the creation of 3D environments for the latest XR hardware devices. Moreover, a growing community is engaged in supporting and expanding these tools. Thus, in many cases game engines are considered a cost-effective and flexible solution for developing XR applications [5]. Nowadays, the ability of game engines to create advanced user experiences and facilitate usage of the latest XR hardware is motivating researchers and practitioners to extend the use of game engines for non-gaming applications [6], such as social and industrial robotics [7]. However, integrating game engines with robotic systems can be a complex task [8]. Moreover, many robotics researchers can have difficulties working with the novel concepts, frameworks, and tools required to use game engines in robotics applications. Therefore, the main objective of this article is to present relevant concepts, tools, and frameworks used or proposed in the literature for enabling the creation of

human–robot XR experiences through general-purpose, accessible, and popular game engines, specifically Unity and Unreal Engine.

1.1. Related Surveys

Recently, several reviews presenting trends and challenges of XR technologies have been proposed for industrial areas; examples include product assembly, maintenance/repair [9], manufacturing training [10], quality control [11], and the automotive industry [12]. While robots can be relevant elements in these applications, human–robot interaction and game engines are not a primary focus of these reviews. Dianatfar et al. [13] presented a narrative review (i.e., one that does not follow a strict methodology to locate articles) of VR and AR solutions in human–robot collaboration. They summarized the main objectives, integration details, and results from 21 research articles proposing VR and AR solutions for collaborative and industrial robots published from 2012 to 2019, of which seven projects used the Unity game engine. Most recently, Suzuki et al. [14] presented a survey and taxonomy of augmented reality and robotics. The focus of [14] was to provide a holistic view that covers broader aspects of the Human–Computer Interaction (HCI) and Human–Robot Interaction (HRI) literature considering any robot-like or actuated system. Finally, Costa et al. [15] presented a systematic literature review of AR applications in HRC in industrial settings. These previous works mostly focus on identifying and classifying hardware devices, interaction modalities, metrics, or applications; to date, little attention has been paid to software aspects. Therefore, the present article attempts to fill this gap, specifically in social and industrial settings. Unlike previous works, this article focuses on reviewing research articles that use game engines as a keystone element in developing XR applications in robotics. The key differences between this article and previous works are summarized in Table 1.

Table 1. Differences between this article and similar works recently presented in the robotics literature.

Reference	Year	Topics Focused on Each Paper	Differences with This Article
[13]	2021	• Review of existing VR/AR solutions in HRC • Solutions are presented in four main categories: operator support, instruction, simulation, and manipulation • Referenced articles published from 2010 to 2019	• Our article presents research questions and a systematic search methodology • Our article presents articles published from 2015 to 2022 • Our article focuses on solutions using game engines as a key technology
[15]	2022	• Categorizes the recent literature on AR for HRC and industrial applications • Identifies the main areas/sectors and employed AR technologies • Highlights key benefits of AR	• Our article reviews papers proposing a VR, MR, and/or AR solution • Our article focuses on solutions using game engines as a key technology • Our article reviews papers for industrial, social, and service applications • Our article highlights reported hardware and software issues
[14]	2022	• Synthesizes and categorizes AR solutions in the following dimensions: approaches to augmenting reality, characteristics of robots, purposes and benefits, classification of presented information, design components and strategies for visual augmentation, interaction techniques and modalities, application domains, and evaluation strategies • The article reviews AR applications presenting any actuated or robot-like hardware (many of which are outside the scope of our review).	• Our article reviews papers proposing VR solutions • Our article focuses on solutions using game engines as a key technology • Our article focuses on papers proposing a solution for industrial, social, and service applications • Our article focuses on software aspects

1.2. Motivation and Contributions

Advances in emergent technologies such as XR are not static. The increasing popularity of XR technologies in recent years has motivated practitioners and researchers to develop new software artifacts to explore the capabilities of new hardware devices. Therefore, keeping this scenario of constant advancement in mind, in addition to the lack

of a comprehensive study focused on the software aspects, there is a need to revisit the state-of-the-art solutions for social, service, and industrial robotics. Therefore, the main contribution of this article is:

The identification of novel and relevant software frameworks and tools for building advanced XR experiences in HRI using popular game engines.

2. Background and Article Organization

According to [16–18], Cyber-Physical Systems (CPSs) can be defined as multi-dimensional and complex systems that integrate computing and networks to govern physical actuators or machines, such as robots. CPSs are a core foundation of Industry 4.0 [19] based on the 3C (computing, communication, and control) approach. The main objective of CPSs is to provide different services (e.g., sensing, information feedback, and dynamic control) to enable reliable, safe, and efficient monitoring and control of physical entities [19]. In this context, digital twins represent an engineering category focusing on virtual models more than 3C capabilities [19]. In addition, ref. [20] demonstrated a proof of concept of digital twin usage in manufacturing. The proposed modules estimate a worker's physical load in a parts-picking scenario and perform dynamic scheduling based on the predicted working progress under ergonomic constraints. In this context, the digital model and digital shadow are two related concepts. On the one hand, a digital model is described in [21] as a representation of reality constructed manually that remains static when reality changes. Therefore, there is no data exchange between the physical and virtual worlds. On the other hand, ref. [21] describes a digital shadow as "an automatically derived model that changes when reality changes". However, in a digital shadow, "there is no automated real-time feedback loop". This is because a digital shadow only provides a one-way data flow from the physical environment to the virtual model [22]. Digital twins go further than digital models and digital shadows, as their results directly trigger or impact reality. Therefore, digital twins provide a two-way data flow from the physical system, both to the virtual system and the other way around [22]. The primary interest of this review is to identify those software artifacts used in HRI applications where the virtual and physical systems can communicate with each other. Therefore, digital models and digital shadows are outside the scope of this article. Most recently, emergent human-centered paradigms, such as Society 5.0 and Industry 5.0, have been envisioned to promote the creation of Human–Cyber-Physical Systems (HCPs), where humans play a central position as "masters" of CPSs [23]. Therefore, the highest right to make decisions always comes from humans, no matter how powerful or sophisticated an intelligent system is. Novel perspectives on these human-centered paradigms have been described in [24,25]. The main focus of the present article is to review those human–robot interaction systems proposed in the literature that use game engines as keystone elements for the creation of complex systems that integrate software frameworks and libraries enabling computing (e.g., object and human recognition), communication between the different modules (e.g., using a socket library or middleware), and control of real robots.

2.1. Game Engines

A game engine is defined in [26] "as a collection of engines for graphics, physics, networking, artificial intelligence (AI), and scripting". A more recent definition is proposed in [6] as an "integrated development environment (IDE) that enables game operators and software developers to create real-time and 3D rendering visualizations". Therefore, at the most basic level, a game engine comprises a rendering side that enables the creation of 3D digital objects (composed of textures, materials, and lighting) and a software development side that allows developers to compile code and execute it on specific devices. While many game engines are available on the market [27], Unity and Unreal Engine in particular have emerged as relevant research tools for non-gaming applications. Below, we briefly review the differences between these game engines (see [28] for a more detailed comparison).

2.1.1. Unity

Unity is a popular and accessible cross-platform game engine enabling the creation of simulated 2D/3D environments and XR applications. The focus of Unity is to provide a general purpose game engine that supports different platforms and users [29]. Unity Projects are composed of *assets* that can be used to create a game or simulation, such as images, audio files, meshes (graphics primitives of Unity), or any other type of file supported in Unity. Assets can be imported to a Unity project using the Unity Package Manager (UPM) or from the Unity Asset Store. A game in Unity is composed of one or multiple scenes (game levels). Each scene contains a hierarchical attachment of basic containers, denoted as GameObjects. The functionality or behavior of GameObjects is defined by built-in components or custom components defined in C# scripts or external plugins [29].

2.1.2. Unreal Engine

Unreal Engine is a popular open and advanced 3D creation platform enabling developers to create immersive experiences and games using photorealistic visuals (i.e., 3D environments that look as realistic as possible). Similar to Unity, projects in Unreal Engine are composed of *assets*. Developers can import assets from other projects or download them from the Unreal assets marketplace. In Unity, Actors are the equivalent of GameObjects in Unreal Engine (i.e., elements that can be placed in the virtual world). Unlike Unity, which mainly uses C#, Unreal Engine uses C++. Therefore, developers require more programming experience in order to create applications. However, applications in Unreal Engine can be performed using a visual scripting tool denoted as *Blueprints*, which can be mixed with C++ scripts.

2.1.3. Paper Organization

Section 3 presents the methodology we followed to perform our systematic search for relevant research articles. Sections 4 and 5 report the study's results. On the one hand, Figure 1 shows a general system architecture summarizing the basic elements presented in the identified articles and an overview of the tools shown in Section 4. In this architecture, a set of sensors collect information from humans and the real-world environment. These sensors include RGB and 3D cameras, game and VR controllers, touch inputs, and wearable devices. Data are sent from the physical system to the virtual system and from the virtual system to the physical system using a communication mechanism (described in Section 4.1). In this review, we focus on those articles that explicitly describe the use of the Unity or Unreal game engines to build or execute virtual systems. A set of interaction tools (described in Section 4.2) process data obtained from sensors and change the real-world environment using robot control tools. Articles presenting this two-way data flow approach between the real and virtual world are the primary focus of this article. On the other hand, Section 5 presents an overview and brief description of applications using similar architectures. Section 6 presents challenges and research opportunities, followed by our conclusions.

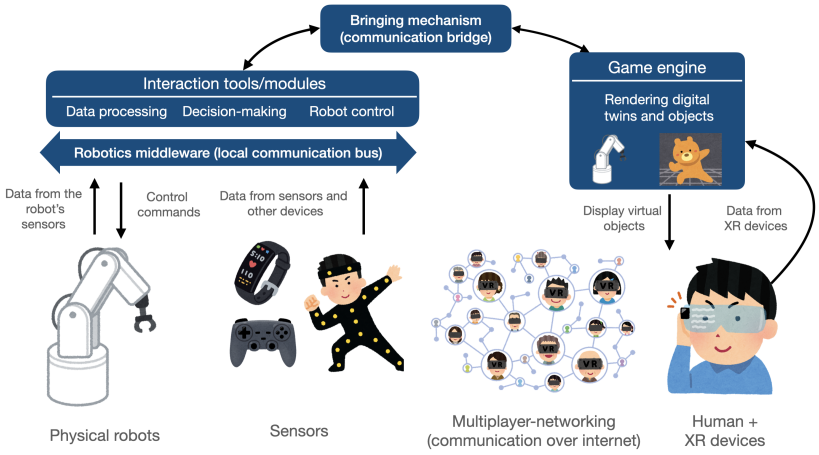

Figure 1. General system architecture of an extended reality application for Human–Robot interaction.

3. Methodology

Scoping reviews are defined in [30] as "a type of knowledge synthesis, follow a systematic approach to map evidence on a topic and identify main concepts, theories, sources, and knowledge gaps". This article follows the guidelines described in [31–33] for performing literature reviews in software engineering and scoping reviews [30]. Consequently, the main steps performed in this review are described below.

3.1. Definition of Research Questions

The research questions (RQs) guiding this study are:

1. **RQ1:** *Which type of XR applications or systems using real robots have been developed using game engines?*
2. **RQ2:** *Which software frameworks or tools are commonly used or have been proposed to build these applications?*
3. **RQ3:** *Which are the reported hardware and software limitations by the authors of these applications?*

3.2. Study Selection, Quality Assessment, and Data Extraction

We limited the search to works published between 2015 and 2022 in order to focus on the most recent developments in this field. The search was performed in October 2022. A pilot search in the ACM and Elsevier databases indicated that after 2015 the interest of the robotics community in the use of game engines with robotic systems began to increase, surpassing a threshold of three papers per year. Keywords composing the search strings were defined using the PICO (Population, Intervention, Comparison, and Outcomes) method suggested in [31]. Several iterations and pilot searches were performed in order to determine an optimum set of primary studies. The strings and advanced settings defined for each database are shown in Table 2.

Table 2. Advanced settings used for each database.

Database	Search String
IEEE Xplore	(Unity OR "Unreal engine") AND Robot AND Human AND (Interaction OR Collaboration) AND ("Virtual reality" OR "Augmented reality" OR "Mixed reality")
ACM Digital Library	[[All: unity] OR [All: "unreal engine"]] AND [Title: robot] AND [Abstract: human] AND [[Abstract: interaction] OR [Abstract: collaboration]]
Springer Link	(Unity OR "Unreal engine") AND Robot AND Human AND (Interaction OR Collaboration) AND ("Virtual reality" OR "Augmented reality" OR "Mixed reality") NOT Medicine NOT Surgery AND (Social OR Industry OR Manufacturing), Content type: Article
Science Direct	All: (Unity OR "Unreal engine"), Title and abstract: Robot AND Human AND (Interaction OR Collaboration OR Control), Subject areas: Engineering, Article type: Research articles
Web of Science	Robot AND Human AND (Interaction OR Collaboration) AND ("Virtual reality" OR "Augmented reality" OR "Mixed reality"), Citation topics: Robotics, and Human-Computer Interaction, Document Types: Article

In this article, the study selection is defined in two steps. In Step 1, we read the title and abstract of each article obtained from the search in the proposed databases. Then, an article was selected as a candidate to be included in the review if it met the following inclusion criteria:

The main focus of the article was to present an XR solution using a social or industrial robot and not other types of hardware artifacts such as drones, swarm robotics, autonomous vehicles, smartphones, intelligent speakers, gloves, or wearable devices.

In Step 2, we skimmed the full text of the articles obtained in Step 1. An article was excluded from the review if it met any of the exclusion criteria:

1. *The article focuses only on simulation (i.e., the robot is only displayed on a monitor) or there is no interaction between the user and a physical robot*
2. *The article is a previous version of a more recent study or system proposing the same software architecture or framework by the same authors*
3. *The article is a short paper, review article, thesis, technical report, or book chapter*
4. *The article is a repeated article in different databases*
5. *The article does not propose a solution using the Unreal Engine or Unity Engine*

Additionally, quality assessment (QA) was defined using the following questions:

1. *Is the article published in a journal that is indexed on Web of Science?*
2. *Was the article presented at a conference in the top twenty Google Scholar list for Robotics, Engineering and Computer Science, Human–Computer Interaction, or Manufacturing and Machines?*
3. *Are the methods, hardware, and software architecture clearly described?*

Articles that did not meet at least one of these quality criteria were excluded from the study. Table 3 shows the number of articles processed in the proposed steps. To answer the proposed research questions, we used an Excel spreadsheet to extract relevant data from each reviewed article. The data extracted from the reviewed articles included year, venue, XR type, application context, software tools, hardware devices, main objective, and reported limitations.

Table 3. Number of studies per database and results after applying inclusion criteria (Step 1), exclusion criteria, and quality assessment criteria (Steps 2 and 3).

Database	Search Results	Articles Included (Step 1)	Articles Excluded (Step 2 and QA)	Articles Finally Included
IEEE Xplore	26	13	9	4
ACM Digital Library	29	6	1	5
Springer Link	137	11	8	3
Science Direct	99	16	8	8
Web of Science	117	23	18	4
Total	408	69	44	24

3.3. Limitations of the Study

Factors limiting the validity of a literature review are described in [33,34]. These potential threats are difficult to eliminate entirely, as they involve human judgment [33]. For example, searches of the same research topic performed by different persons can end up with different sets of articles and extracted data due to researcher bias [33,35]. To reduce these problems, in this study researchers experienced in different areas of robotics, such as social and physical HRI, robot manipulation and control, and artificial intelligence, were involved in validation of the extracted data and conclusions.

4. Software Tools for Developing XR Applications in HRI

This section describes commonly used software tools reported to be integrated or used in the reviewed articles. We classified the identified software tools based on the 3C paradigm. However, we considered tools enabling computation (sensing and processing of data) and control of physical systems in the category of interaction. Figure 1 shows

a general system architecture summarizing the main elements presented in the robotic systems reviewed in this article. In this architecture, a set of sensors collect information from interaction with the environment. Examples of data collected from sensors attached to robots include joint angles, physical interactions (from touch sensors), and forces, while examples of data obtained from humans include heart rate, inertial data, inputs from game controllers, and speech.

4.1. Communication Tools

Enabling the exchange of information produced in virtual worlds and generated by humans and robots in the real world is not a trivial task. In order for it to be successful, users must use different communication or networking tools. Below, we briefly describe the communication tools most frequently used in the reviewed articles.

4.1.1. Robotic Middleware and Bridging Mechanism

Communicating with game engines and external robotic modules often requires interprocess communication using socket connections. However, managing socket connections can become a cumbersome task in complex software architectures. Therefore, using robotic middleware is often recommended in order to reduce the management complexity and improve the reusability of complex robotic software architectures. We observed that the Robot Operating System (ROS) version 1.0 was the most popular middleware solution used by the authors of the reviewed articles. However, the connection between ROS1 modules and virtual environments developed in Unity and Unreal cannot be made directly due to differences in dependencies, programming languages, communication protocols, and/or operating systems. In order to deal with this issue, bridging mechanisms or modules are often used. These modules generally translate and transfer data from a game engine to a non-compatible robotic middleware that uses another message formatting or communication protocol. In the case of ROS, the most popular solution enabling the connection of non-ROS programs to an ROS network is rosbridge_suite [36], which provides a solution based on JavaScript object notation (JSON) and Websockets. However, ref. [37] reported performance limitations of rosbridge_suite when transferring large sensory messages (e.g., images and point cloud data), which specifically affects transfer speed. In this direction, ROS# [38] has been widely used by the reviewed articles for communicating ROS with Unity. This framework provides an ROS Bridge Client for C# applications, a Unified Robot Description Format (URDF) file parser that enables developers to import robot models to Unity, and a message generation tool for sending and receiving ROS messages from Unity. To overcome the performance limitations of rosbridge_suite, alternatives based on TCP/IP sockets instead of Websockets, such as [37], have been proposed. More recently, the TCP connector [39] framework for Unity has been released by the developers of ROS# (sharing many of the same tools) as an official Unity asset. However, none of the reviewed articles has reported the integration of this framework. Another alternative explored by [40,41] proposes alternatives for connecting Unity with external robotic systems using a middleware solution based on ZeroMQ [42]. Unlike conventional sockets, which present a synchronous interface, ZeroMQ sockets are asynchronous. Moreover, as proven in [40,41], tools based on ZeroMQ can be used to overcome the communication performance limitations of rosbridge_suite. Most recently, ros2-for-unit (https://github.com/RobotecAI/ros2-for-unity [access date: 12 December 2022]) has been proposed as an unbridged alternative to connect ROS2 and Unity. The authors of this tool state that it provides considerably lower latency than solutions using rosbridge_suite.

4.1.2. Multiplayer Networking

The use of game engines for developing XR experiences in HRI applications can benefit from the technology used to create online multiplayer games. In this type of game, users are not restricted to being located in the same local network, and can interact with other players around the world. In this context, game engines can enable the relatively easy integration

of high-level multiplayer networking tools. These tools can handle communication and synchronization of objects presented in virtual environments and user authentication. Examples of these tools include Photon [43], Mirror [44], and the recently launched Netcode for GameObjects [45]. However, very few works, among which are included [37,46,47] have used this technology to enable remote and/or multi-party HRI in virtual environments.

4.2. Interaction Tools

Several works identified in this article integrate one or more software libraries and frameworks to detect human actions or states of the environment. In this context, ref. [48] integrated TouchScript [49], a multi-touch library for Unity, is used to enable multi-touch gesture recognition in a tangible user interface (TUI). Moreover, ref. [48] integrated Point Cloud Library (PCL) [50] to detect objects from a table using the Random Sample Consensus method [51]. Zhou et al. [52] used the PointNet [53] neural network library to classify objects from point cloud data. Examples of applications enabling PointNet include classification, part segmentation, and scene semantic parsing. In the case of AR projects, the Vuforia Engine SDK [54] has been used in a number of the reviewed articles. This software tool enables developers to add advanced computer vision functionalities to build AR applications for mobile devices and AR glasses, such as recognizing and tracking objects by shape using digital 3D models and attaching AR content to flat images or objects with cylindrical bodies, among others. An alternative solution for creating AR applications on mobile devices used in [55] is Google's ARCore library for Unity [56]. In the case of AR glasses, specifically Hololens, the Mixed Reality Toolkit [57] is generally used for AR/MR development in Unity. Additionally, ref. [58] used the ZXing [59] barcode image processing library to link the world coordinates of a virtual environment with the physical world. For skeleton tracking, ref. [60] implemented the NuiTrack 3D tracking framework [61], which provides cross-platform solutions for body motion analysis. This framework is compatible with several depth sensors, such as Kinect Azure, Intel Realsense, and Asus Xtion. One of the most popular libraries used in the reviewed articles is the OpenCV [62] image-processing library, which has become quasi-standard in many computer vision applications. OpenCV is often used as a dependency with other software tools using computer vision algorithms, such as the Find_Object_2D [63] ROS package, which is used in [60] to perform feature extraction from images of objects. For motion planning and control of robots, many reviewed articles use the MoveIt Motion Planning Framework [64]. Other tools used in the reviewed articles include IAI Kinect [65] (for bridge Kinect and ROS), RobCog-IAI [66] (to develop robot applications using Unreal Engine), and Newton VR (to manipulate virtual objects using tracking controllers) [67]. Basic specifications, descriptions, and capabilities of the tools mentioned in this section are shown in Table 4.

Table 4. Specifications and capabilities of identified interaction tools in the reviewed articles.

Tool	Programming Languages	General Description and Main Capabilities/Functionalities
TouchScript	C#	An open-source framework that enables the use of basic gestures (e.g., press, release, tap, long press, flick, pinch/scale/rotate) in devices with touch input
Point Cloud Library (PCL)	C++	An open-source library for 2D/3D image and point cloud processing. Some of its main modules include visualization, segmentation, and registration of point clouds.
PointNet	Python	A deep net architecture and library written on top of TensorFlow. It can be used to reason about 3D geometric data (e.g., point clouds and meshes). Its functionalities include part segmentation, object classification, and scene semantic parsing.

Table 4. *Cont.*

Tool	Programming Languages	General Description and Main Capabilities/Functionalities
Vuforia Engine SDK	C++, Objective-C++, Java, and C#	A framework for building Augmented Reality applications in Android, iOS, and Windows applications able to be executed on mobile devices and AR glasses. It provides free, academic, and premium plans.
Google's ARCore	Kotlin/Java, Unity/C#, Unreal/C++	A framework for building Augmented Reality applications in Android, iOS, Unreal, and Unity. It uses motion tracking, environmental understanding, and light estimation to integrate virtual models with the real world
Mixed Reality Toolkit	C#	Is a cross-platform framework providing a set of components and features for accelerating Mixed and Augmented reality application development. Its functionalities include eye and hand tracking, spatial awareness, speech dictation, and XR device and game control manager.
ZXing	Java	An open-source library for 1D/2D barcode image processing. Its functionalities include analysis and extraction of information from images containing barcodes or QR codes
NuiTrack	C++, C#, and Python	A 3D body tracking middleware. This framework can interpret depth maps as 3D point clouds, detect floor planes and background objects, detect and track persons, perform skeleton and hand tracking, and perform face analysis.
OpenCV	C++/C, Python, and Java	A cross-platform and general-purpose real-time computer vision library. Its functionalities include image acquisition from RGB cameras, image segmentation and filtering, object recognition, camera calibration, and changing image color spaces, among many others.
Find_Object_2D	C++	A ROS-based package using a Qt interface that implements different feature detectors and descriptors for calculating the 3D position of objects using OpenCV
MoveIt	C++ and Python	Robot motion planning framework enabling motion planning, manipulation, collision checking, 3D perception, kinematics, control, and navigation
IAI Kinect	C++	Collection of tools for enabling the use of Kinect with ROS using libfreenect2 (a driver for Kinect)
RobCog-IAI	C++	Provides Unreal Engine plugins for building applications for robotics
Newton VR	C#	A VR interaction system enabling to pick up, throw, and use objects using tracked controllers

5. Applications in Human–Robot Interaction

This section briefly describes the main contributions of the reviewed articles. We have classified these articles into four groups representing the main context or application area; therefore, this section answers RQ1. These articles are summarized in Tables 5–8.

Table 5. Articles presenting XR solutions for social robotics.

Article			Hardware Integrated		Software Tools Integrated	
Year	Reference	XR Type	Robots	Devices	Communication	Interaction
2019	[68]	VR	Pepper	HTC VIVE	TCP/IP	NAO SDK
2021	[60]	VR	QTRobot	Oculus Rift, iphone	ROS and ROS bridge	NuiTrack, Find_Object_2D
2022	[69]	VR	NAO	Oculus Rift	ROS, ROS#	NAO SDK

5.1. Social Robotics

The results of our review indicate that the use of game engines to build HRI applications using XR technologies for service and social robotics remains in its initial stages. In this context, a common application is to use a VR environment to replace a real robot, as presented in [70]. However, few works integrate real social robots with virtual environ-

ments. An exception is [68], which uses a VR environment to investigate the differences between a real Pepper robot and its virtual representation by using both a virtual and a real robot simultaneously. For this, Client/Server communication using traditional TPC/IP sockets in a local area network is used to connect Unity with Choregraphe [71] (the official visual programming environment [72] and simulation platform for Pepper and NAO social robots). The results of [68] suggest that users may prefer closer interaction with a physical robot than its virtual representation. The authors of [68] infer that this difference can be influenced by the potential discomfort that can be produced by a VR environment with technical limitations, such as a limited field of view, low visual fidelity, or a low-resolution display. Another exception is using VR environments to remotely control social robots. In this context, refs. [60,69] have proposed ROS-based systems for controlling NAO and QTRobot.

Table 6. Articles presenting XR solutions for programming robots using game engines.

Article			Hardware Integrated		Software Tools Integrated	
Year	**Reference**	**XR Type**	**Robots**	**Devices**	**Communication**	**Interaction**
2019	[48]	AR	UR5	Kinect 2, projector	ROS, ROS#	TouchScript, Point Cloud Library
2019	[73]	AR	GripperBot, CamBot, Armbot	Oculus Rift and IR-LED Sensors, ZED stereocamera	ROS, ROS#	N.A
2020	[74]	MR	KUKA iiwa, UR10e	HoloLens	ROS (no details on how Unity and ROS are connected)	MoveIt, Mixed Reality Toolkit
2020	[55]	MR/AR	OpenManipulator-X	Smartphone	Bluetooth	Google's ARCore
2021	[75]	AR	UR5 and ABB IRB 2600	HoloLens 2	ROS, ROS#, TCP/IP sockets	N.A.
2021	[58]	AR	UR10	HoloLens	ROS, ROS#	ZXing, Mixed Reality Toolkit, MoveIt

5.2. End User Programming of Industrial Robots

Robot programming is an emergent area where game engines and XR technologies are implemented for creating HRI systems. In these applications, the reduction of training effort available when using intuitive, simple, and usable interfaces is fundamental. In this context, ref. [73] have presented GhostAR. This time-space editor combines the Oculus Rift and the ZED Dual Cameras to enable users to program and edit the actions of robots through Programming by Demonstration (PbD). This system can display "ghosts", images representing a timeline of captured human motions. These "ghosts" are used as time-space references, which can be used to edit HRC tasks using a virtual avatar. PbD solutions using the HoloLens have been explored using AR [58] and MR [74] methods, providing promising results compared to conventional programming through teaching pendant and kinesthetic programming. In this context, ref. [75] have proposed a markerless solution that records and translates the operator's hand movements into robot motions and commands using Hololens 2. However, HoloLens is considered an expensive device [76–78]. In contrast, smartphone-based AR applications such as [55] can represent a more cost-effective solution. The authors of [75] claim that their proposed approach "is computationally lightweight compared to other Learning from Demonstration techniques, such as Machine Learning approaches". However, they claim that this solution may not be suitable for applications that require high accuracy in robot motions. A projection-augmented tabletop interface, denoted as PATI, was proposed in [48] to overcome the perception limits (visual tracking and object referencing) of PbD. In PATI, the task-level specifications defined in the TUI are translated to robot commands and motion plans. For this, the PATI system can detect touch gestures and track objects in the working environment. The authors of PATI claim that their

system can be learned in a shorter period of time than state-of-the-art PbD systemsm, and can enable users to be more efficient, producing less physical and mental workload.

Table 7. Articles presenting XR solutions for the teleoperation of industrial robots using game engines.

Article			Hardware Integrated		Software Tools Integrated	
Year	Reference	XR Type	Robots	Devices	Communication	Interaction
2020	[79]	AR	Kuka KR6 r900	Realsense D435, Gamepad, HoloLens	ROS and ROS bridge	Mixed Reality Toolkit
2020	[52]	VR	Baxter	Kinect, HTC VIVE	ROS, ROS2Unity, ROS#	PointNet, IAI Kinect, libfreenect2, PCL
2021	[47]	VR	COMAU Dual Arm	Oculus Rift	ROS, ROS#, Mirror	Moveit, Open Motion Planning Library
2021	[80]	VR	UR5	Pro, ZED-mini, Intel Realsense, HTC VIVE	ROS and ROS bridge, TCP sockets	RobCog-IAI
2022	[81]	MR	UR5	Camera, Kinect, HTC VIVE	ROS and ROS bridge	OpenCV

5.3. Teleoperation

VR and AR systems are suitable solutions for enabling remote control of robots in scenarios that are inaccessible or dangerous for humans. Relevant considerations in the design of effective teleoperation systems highlighted by the reviewed articles include intuitiveness, low latency, and situational awareness [47,52]. In this context, ref. [47] propose an intuitive VR interface for enabling multiple users simultaneously immersed in a synchronized VR scene to define a robot's positions and actions. A solution to increase the human operators' situational awareness in teleoperation tasks is proposed in [52]. For this, the authors of [52] present the Telerobotic Operation based on an Auto-reconstructed Remote Scene (TOARS) algorithm, which uses deep learning methods to detect objects and their physical properties from point cloud information generated by a Kinect v2. This information is subsequently rendered as virtual objects with physical properties (e.g., friction and weight) in the Unity-based simulation to provide a more immersive interaction. In addition, the authors provide an alternative for reducing communication latency produced by rosbridge_suite. For this, they reduce the amount of transferred data over the network generated by Kinect v2. Moreover, a TCP/IP connection between Unity and ROS is implemented to improve the communication performance in comparison with ROS#. An AR solution using gamepads for the teleoperation of industrial robots is proposed and evaluated in [79]. Results from usability testing performed in [79] suggest that gamepads can provide higher usability for teleoperation tasks than conventional teaching pendant devices, regardless of the user's previous experience in robotics and/or AR technology. Unlike most of the reviewed articles, which use Unity as the main game engine platform, ref. [80] proposes a VR solution for robot teleoperation developed with Unreal Engine version 4 for high-fidelity visualization, denoted as Vicarios. The Vicarios framework aims to improve situational awareness and user performance in remote robot manipulation tasks by implementing a method denoted as viewpoint-independent motion mapping. In this approach, operators can choose between different viewpoints without worrying about remapping their motions to each viewpoint. Vicarios uses the robcog-iai [66] tools to import 3D models of robots using the URDF, then connects the Unreal Engine to ROS. Two MR-based hybrid 3D/2D visualization paradigms, denoted as MR-3DSV (integrating 3D stereoscopic vision and monocular RGB cameras) and MR-3DPC (integrating 3D point cloud and monocular RGB cameras), are proposed in [81]. These paradigms aim to provide more immersive and intuitive teleoperation for non-skilled operators. The results from [81] indicate that using hybrid 3D/2D vision paradigms can reduce the training effort, completion time, and cognitive workload compared with

conventional 2D-based teleoperation methods (i.e., only using a 2D camera stream to perceive the robot space). Finally, ROS Reality [82] is a VR Framework and ROS package enabling the teleoperation of ROS-enabled robots using Unity-compatible VR headsets. This framework provides different modules, including WebSocket communication with ROS, a URDF parser that allows 3D models of ROS-based robots to be imported to Unity, an RGB camera, and a point cloud and inverse kinematic status visualizer. Experimental validation of the ROS Reality framework was carried out in a pilot study that compared the efficacy against the kinesthetic handling of the robot. The results suggested that the proposed VR architecture was less effective than direct manipulation of the robots. However, the results additionally indicated that teleoperation solutions using VR environments can be more useful than directly manipulating a real robot when complex movements of the robot's joints are required. The source code of ROS Reality is available on GitHub (https://github.com/h2r/ros_reality, accessed on 10 October 2022).

Table 8. Articles presenting XR solutions supporting Human–Robot collaboration tasks.

Article			Hardware Integrated		Software Tools Integrated	
Year	**Reference**	**XR Type**	**Robots**	**Devices**	**Communication**	**Interaction**
2018	[83]	AR	Turtlebot	Tablet	ROS (no details on how Unity and ROS are connected)	Vuforia SDK
2020	[46]	AR	COMAU AURA, COMAU Racer5 HoloLens	Smartwatch, Projector, Camera	ROS Bridge and Mirror	Microsoft Speech recognition library
2020	[84]	AR	KUKA KR6 R7000 Sixx	HoloLens	ROS and ROS bridge #,	Vuforia SDK
2021	[85]	VR	Franka Emika	N.A	ROS, ROS#	N.A.
2021	[86]	AR	Turtlebot2	Tablet	ROS, ROS#	N.A.
2021	[37]	VR/AR	Turtlebot2, Turtlebot3, TIAGo, among others	Oculus Rift	ROS, ROS Bridge, TCP/IP, Photon	NewtonVR
2022	[87]	VR	Scitos G5 mobile robot	HoloLens 2, Azure Kinect	ROS, ROS#	Mixed Reality Toolkit
2022	[88]	MR/AR	ABB industrial robot	HoloLens	FB network	N.A
2022	[89]	VR	Universal Robot	Xsens motion, HTC VIVE	ROS, ROS#	N.A
2022	[90]	AR	Mobile dual-arm platform	HoloLens 2	ROS, ROS#	Mixed Reality Toolkit

5.4. Human–Robot Collaboration

The creation of XR systems enabling robots to perform collaborative tasks with humans has been widely explored. In this context, ref. [85] presented a set of techniques for ad hoc HRC (i.e., collaborative activities where the robot works without pre-planned actions) using digital tabletops and mixed reality. However, restrictions presented by the COVID-19 pandemic constrained the experimental validation of the system in a VR simulation. A solution that links AR and eye-tracking technologies to enable 3D object detection in a collaborative way was presented in [87]. A key novelty that the authors of [87] presented is the use of gaze information to help detect and segment unknown objects. For this, they used the HoloLens 2 AR device, which has a built-in eye tracker over Unity, and for which information is connected with ROS using ROS#. The authors claimed to be the first to use AR in an HRC setting to segment unknown 3D objects without deep learning techniques, and their system was able deal with cases where existing deep neural learning techniques have failed. Other related works using Hololens and ROS (connected by ROS#) to facilitate HRC activities have been described in [84,91]. A novel approach using Hololens for empowering human-centric assembly (HCA) by combining a digital twin [92] developed in a Java-based 3D environment with AR methods (executed in Unity) and event-driven function blocks [93] has recently been proposed by [88]. Validation of their system was

centered on performance evaluation in an engine assembly case. A framework denoted as Operator Support Module (OSM) was presented in [46] for facilitating collaboration by providing cognitive support to operators using AR technologies, voice commands, and gestures. The authors stated that OSM can recognize objects and provide feedback to users, e.g., when a wrong object is manipulated or when a step in the assembly task is completed. Due to the expensive deep learning methods used in [46], the system was distributed on different computers. A module in Unity denoted OSM Controller was proposed to handle communication between the different elements in the system. This module connects information generated by Unity with ROS (using rosbridge) and Unity with wearable and smart devices (using Mirror). We observe that studying ergonomic and performance aspects in HRC industrial tasks using XR systems is becoming an emergent topic. In this context, ref. [89] proposed a methodology to determine the accuracy and quality of a human motion capture system integrating an inertial measurement unit (IMU) and Vive-based HTC during physical human–robot interaction. For this, they created a digital twin of the actual system in which human and robot motions are replicated in real-time in the visual environment. To transfer the actual robot motion to the digital robot model, they used ROS#. An effort to understand the variables affecting the trust involved in HRC using AR technologies was presented in [83]. The presented results suggest that context awareness and human safety are key variables affecting trust in HRC scenarios. Though presented in [83] was performed with a Turtlebot mobile robot, the authors stated that similar results could be obtained with industrial cobots, which are heavy and strong. However, their authors note that further research involving a real industrial cobot is needed in order to better understand the factors affecting trust in cobots. Most recently, ref. [90] proposed a software suite for supporting the operator's interaction with mobile and industrial robot co-workers. This suite proposes a solution for easy programming, supporting autonomous operations, assisting operators in assembly processes, and enhancing safety awareness and resilience of production processes. The experimental evaluation proposed in [90] suggests that the proposed system can reduce the time required by operators to recover from operational failures and safety violations. AR for Robots Collaborating with a Human (ARROCH) [86] is a system enabling communication between humans and multiple robots. With ARROCH, an operator can use a tablet to visualize the robots' current states and planned actions (intentions). Finally, the SIGverse [37] cloud-based VR platform, built on top of Unity, proposes solutions for performing different HRI experiments. With SIGverse, one or more users can simultaneously interact with virtual robots at the same time using the Photon Unity Networking (PUN) asset for Unity, which provides multiplayer networking features. A bridging mechanism is used to ensure communication between VR scenes created in Unity and ROS modules. This bridging mechanism uses WebSockets and the JSON format to send small-size data from Unity to ROS, a solution based on TCP/IP sockets, and uses binary JavaScript object notation (BSON) format to send large data messages. The user cases reported in [37] include robotic competitions, collaborative clean-up of a room using human pointing gestures, evaluation of the subjective quality of HRI, and motion learning by demonstration. SIGVerse is available on GitHub (https://github.com/SIGVerse, accessed on 10 October 2022), and its documentation can be found on its website (http://www.sigverse.org/wiki/en/, accessed on 10 October 2022).

6. Challenges and Future Opportunities

This section summarizes a set of hardware and software issues reported by the authors of the reviewed articles. Therefore, this section answers research question RQ3.

Ergonomics and comfort issues of popular XR headsets have been repeatedly exposed as one of the most relevant limitations of the systems proposed in the reviewed articles. In this context, ref. [58] noted that HoloLens, the most popular AR device used in the reviewed articles, is considered uncomfortable after long usage periods due to its size and weight. Similar results have been presented in experiments conducted in [79]. Experiments presented in [91] indicate that the ergonomics of the HoloLens headset represent a relevant

drawback, making AR solutions ineligible for full-scale usage in working environments despite the advanced features and functionalities they can provide. Aivaliotis et al. note that almost every VR and AR headset on the market is unsuitable for a regular eight-hour workday [90]. Moreover, a notable percentage of the population suffers cybersickness after long sessions using VR/AR headsets [94]. We observed that few reviewed articles using these XR devices reported any countermeasures provided to reduce cybersickness or build tolerance. Exceptions are [37,95], where an analysis of the elements or factors causing cybersickness (e.g., when motions in a virtual world are too fast compared with reality) is performed, with the offending element modified or deleted to improve performance and user experience. Furthermore, many VR/AR systems require the use of expensive hardware and computational components and complex software tools. For example, AR devices such as Hololens 2 cost at least USD 3500, representing an expense that many researchers, sectors, or users cannot afford [76–78]. Moreover, many VR/AR solutions require the acquisition of mid-range or high-end computers to run XR environments and perceptual and robot control algorithms. In this respect, the availability of more accessible hardware devices such as Oculus and more efficient algorithms such as the mediapipe framework can reduce the need to acquire high-performance graphics cards. On the other hand, ROS has become a popular solution in robotics research and industrial applications. However, "it requires significant training and experience before one can use it to competently program robot tasks" [48]. We observe that many of the projects presented in this review require the use of both Linux (for robot control) and Windows (for executing the virtual environment) machines at the same time. These machines must communicate using the tools described in Section 4.1. In cases where only a Windows computer is available, it is necessary to port the robotic code executed in Linux. This code can be embedded in Unity as C# scripts, as suggested by [37], or executed as external modules (e.g., written in Python or C++). However, this can be a complex and time-consuming task. Therefore, researchers and practitioners must discuss the suitability of these approaches [37] based on the needs, experience, and economic/computational resources of the end users. For robots providing cross-platform software development kits (SDK), such as NAO and Pepper robots, relying on user-friendly and platform-independent communication tools based on lightweight and portable libraries can be a suitable and user-friendly alternative to ROS in certain cases. An example in this direction is presented in [96], which proposes a cross-platform system and low-cost VR alternative for industrial robots using the Message Queuing Telemetry Transport (MQTT) protocol.

According to [97], sustainability aspects (e.g., computational performance and power consumption), which is a Society 5.0 and Industry 5.0 [24] main concern, are rarely contemplated by the robotics research community. Additionally, cost and resource consumption can be particularly relevant when developing robotic systems for use outside laboratories and industrial applications [97]. As suggested in [37,48,58], lowering the economic and technical barriers of XR systems is relevant for promoting their adoption in custom contexts and exploring the potential benefits that robots and XR technologies can provide together. Moreover, the lack of reliability and ease-of-use aspects in many cases is used by industry to justify the hesitation and lack of acceptance of XR solutions for use on larger scales [58].

In this review, we observed that the Unity game engine was the most commonly used platform in the reviewed articles. The only exception was [80], which used Unreal Engine 4. One factor that can motivate robotics researchers to use Unity over Unreal engine is that Unity presents a lower learning curve and requires fewer programming skills to build virtual worlds. The very recent release of Unreal Engine 5 promises photorealistic visuals, which can be used by developers and researchers to create immersive experiences. However, usable tools and frameworks that further reduce the effort required to produce XR applications that integrate physical robots remain required in order to expand the popularity of the Unreal engine in the HRI community.

7. Conclusions

This review article has identified commonly used and relevant software frameworks and tools used by the robotics community to integrate physical robotic platforms with virtual elements generated by game engines, specifically, the Unity and Unreal Engines. We have presented and classified these software tools as networking (including robotic middleware and multiplayer networking) and interaction tools. Additionally, we have presented the main contributions of the reviewed articles and classified their application contexts into four main groups: social robotics, teleoperation, end-user programming, and HRC. We observe that using these game engines along with robotic middleware and multiplayer networking frameworks is becoming a trend in the creation of XR applications in HRI for industrial settings. However, its use for social robotics remains in the initial stages.

The world is passing through a paradigm change towards Society 5.0 and Industry 5.0, and XR technologies are often considered keystone elements of these paradigms. However, the articles reviewed in this work suggest that the acceptance and success of these technologies in HRI depends on both the efficiency that the proposed systems are able to achieve and on the human, economic, and sustainability factors of their software and hardware elements.

Author Contributions: Conceptualization, E.C.; methodology, E.C.; validation, E.C., S.I. and I.G.R.-A.; formal analysis, E.C.; investigation, E.C.; resources, E.C. and S.I.; data curation, E.C. and S.I.; writing—original draft preparation, E.C. and S.I.; writing—review and editing, I.G.R.-A.; visualization, E.C.; supervision, I.G.R.-A.; project administration, I.G.R.-A.; funding acquisition, I.G.R.-A. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

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. Kwok, A.O.; Koh, S.G. COVID-19 and extended reality (XR). *Curr. Issues Tour.* **2021**, *24*, 1935–1940. [CrossRef]
2. Doolani, S.; Wessels, C.; Kanal, V.; Sevastopoulos, C.; Jaiswal, A.; Nambiappan, H.; Makedon, F. A review of extended reality (xr) technologies for manufacturing training. *Technologies* **2020**, *8*, 77. [CrossRef]
3. Bogaerts, B.; Sels, S.; Vanlanduit, S.; Penne, R. Connecting the CoppeliaSim robotics simulator to virtual reality. *SoftwareX* **2020**, *11*, 100426. [CrossRef]
4. Topini, A.; Sansom, W.; Secciani, N.; Bartalucci, L.; Ridolfi, A.; Allotta, B. Variable admittance control of a hand exoskeleton for virtual reality-based rehabilitation tasks. *Front. Neurorobot.* **2021**, *15*, 188. [CrossRef] [PubMed]
5. Nguyen, V.T.; Dang, T. Setting up Virtual Reality and Augmented Reality Learning Environment in Unity. In Proceedings of the 2017 IEEE International Symposium on Mixed and Augmented Reality (ISMAR-Adjunct), Recife, Brazil, 9–13 November 2017; pp. 315–320. [CrossRef]
6. Morse, C. *Gaming Engines: Unity, Unreal, and Interactive 3D Spaces*; Taylor & Francis: London, UK, 2021.
7. Bartneck, C.; Soucy, M.; Fleuret, K.; Sandoval, E.B. The robot engine—Making the unity 3D game engine work for HRI. In Proceedings of the 2015 24th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN), Kobe, Japan, 31 August–4 September 2015; IEEE: Piscataway Township, NJ, USA, 2015; pp. 431–437.
8. De Melo, M.S.P.; da Silva Neto, J.G.; Da Silva, P.J.L.; Teixeira, J.M.X.N.; Teichrieb, V. Analysis and comparison of robotics 3d simulators. In Proceedings of the 2019 21st Symposium on Virtual and Augmented Reality (SVR), Rio de Janeiro, Brazil, 28–31 October 2019; IEEE: Piscataway Township, NJ, USA, 2019; pp. 242–251.
9. Eswaran, M.; Gulivindala, A.K.; Inkulu, A.K.; Bahubalendruni, M. Augmented reality-based guidance in product assembly and maintenance/repair perspective: A state of the art review on challenges and opportunities. *Expert Syst. Appl.* **2023**, *213*, 1–18. [CrossRef]
10. Zhang, W.; Wang, Z. Theory and Practice of VR/AR in K-12 Science Education—A Systematic Review. *Sustainability* **2021**, *13*, 12646. [CrossRef]
11. Ho, P.T.; Albajez, J.A.; Santolaria, J.; Yagüe-Fabra, J.A. Study of Augmented Reality Based Manufacturing for Further Integration of Quality Control 4.0: A Systematic Literature Review. *Appl. Sci.* **2022**, *12*, 1961. [CrossRef]

12. Boboc, R.G.; Gîrbacia, F.; Butilă, E.V. The application of augmented reality in the automotive industry: A systematic literature review. *Appl. Sci.* **2020**, *10*, 4259. [CrossRef]
13. Dianatfar, M.; Latokartano, J.; Lanz, M. Review on existing VR/AR solutions in human–robot collaboration. *Procedia CIRP* **2021**, *97*, 407–411. [CrossRef]
14. Suzuki, R.; Karim, A.; Xia, T.; Hedayati, H.; Marquardt, N. Augmented Reality and Robotics: A Survey and Taxonomy for AR-enhanced Human-Robot Interaction and Robotic Interfaces. In Proceedings of the CHI Conference on Human Factors in Computing Systems 2022, New Orleans, LA, USA, 30 April–5 May 2022; pp. 1–33.
15. Costa, G.d.M.; Petry, M.R.; Moreira, A.P. Augmented Reality for Human &;Robot Collaboration and Cooperation in Industrial Applications: A Systematic Literature Review. *Sensors* **2022**, *22*, 2725. [CrossRef]
16. Xie, J.; Liu, S.; Wang, X. Framework for a closed-loop cooperative human Cyber-Physical System for the mining industry driven by VR and AR: MHCPS. *Comput. Ind. Eng.* **2022**, *168*, 108050. [CrossRef]
17. Sonkoly, B.; Haja, D.; Németh, B.; Szalay, M.; Czentye, J.; Szabó, R.; Ullah, R.; Kim, B.S.; Toka, L. Scalable edge cloud platforms for IoT services. *J. Netw. Comput. Appl.* **2020**, *170*, 102785. [CrossRef]
18. Zanero, S. Cyber-Physical Systems. *Computer* **2017**, *50*, 14–16. [CrossRef]
19. Tao, F.; Qi, Q.; Wang, L.; Nee, A. Digital Twins and Cyber–Physical Systems toward Smart Manufacturing and Industry 4.0: Correlation and Comparison. *Engineering* **2019**, *5*, 653–661. [CrossRef]
20. Maruyama, T.; Ueshiba, T.; Tada, M.; Toda, H.; Endo, Y.; Domae, Y.; Nakabo, Y.; Mori, T.; Suita, K. Digital Twin-Driven Human Robot Collaboration Using a Digital Human. *Sensors* **2021**, *21*, 8266. [CrossRef] [PubMed]
21. van der Aalst, W.M.; Hinz, O.; Weinhardt, C. Resilient digital twins. *Bus. Inf. Syst. Eng.* **2021**, *63*, 615–619. [CrossRef]
22. Sepasgozar, S.M. Differentiating digital twin from digital shadow: Elucidating a paradigm shift to expedite a smart, sustainable built environment. *Buildings* **2021**, *11*, 151. [CrossRef]
23. Zhou, J.; Zhou, Y.; Wang, B.; Zang, J. Human–cyber–physical systems (HCPSs) in the context of new-generation intelligent manufacturing. *Engineering* **2019**, *5*, 624–636. [CrossRef]
24. Coronado, E.; Kiyokawa, T.; Ricardez, G.A.G.; Ramirez-Alpizar, I.G.; Venture, G.; Yamanobe, N. Evaluating quality in human-robot interaction: A systematic search and classification of performance and human-centered factors, measures and metrics towards an industry 5.0. *J. Manuf. Syst.* **2022**, *63*, 392–410. [CrossRef]
25. Huang, S.; Wang, B.; Li, X.; Zheng, P.; Mourtzis, D.; Wang, L. Industry 5.0 and Society 5.0—Comparison, complementation and co-evolution. *J. Manuf. Syst.* **2022**, *64*, 424–428. [CrossRef]
26. Eberly, D. *3D Game Engine Design: A Practical Approach to Real-Time Computer Graphics*; CRC Press: Boca Raton, FL, USA, 2006.
27. Gregory, J. *Game Engine Architecture*; AK Peters/CRC Press: Boca Raton, FL, USA, 2018.
28. Dickson, P.E.; Block, J.E.; Echevarria, G.N.; Keenan, K.C. An experience-based comparison of unity and unreal for a stand-alone 3D game development course. In Proceedings of the 2017 ACM Conference on Innovation and Technology in Computer Science Education, Bologna, Italy, 3–5 July 2017; pp. 70–75.
29. 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.
30. Tricco, A.C.; Lillie, E.; Zarin, W.; O'Brien, K.K.; Colquhoun, H.; Levac, D.; Moher, D.; Peters, M.D.; Horsley, T.; Weeks, L.; et al. PRISMA extension for scoping reviews (PRISMA-ScR): Checklist and explanation. *Ann. Intern. Med.* **2018**, *169*, 467–473. [CrossRef] [PubMed]
31. Budgen, D.; Brereton, P. Performing systematic literature reviews in software engineering. In Proceedings of the International conference on Software engineering. *Assoc. Comput. Mach.* **2006**, 1051–1052. [CrossRef]
32. Kitchenham, B. *Procedures for Performing Systematic Reviews*; Technical Report; Keele University: Newcastle, UK, 2004.
33. Petersen, K.; Vakkalanka, S.; Kuzniarz, L. Guidelines for conducting systematic mapping studies in software engineering: An update. *Inf. Softw. Technol.* **2015**, *64*, 1–18. [CrossRef]
34. Keele, S. *Guidelines for Performing Systematic Literature Reviews in Software Engineering*; Technical Report, EBSE Technical Report; Elsevier: Amsterdam, The Netherlands, 2007.
35. Wohlin, C.; Runeson, P.; Neto, P.A.d.M.S.; Engström, E.; do Carmo Machado, I.; De Almeida, E.S. On the reliability of mapping studies in software engineering. *J. Syst. Softw.* **2013**, *86*, 2594–2610. [CrossRef]
36. Rosbridge Suite. 2022. Available online: http://wiki.ros.org/rosbridge/_suite (accessed on 10 October 2022).
37. Inamura, T.; Mizuchi, Y. SIGVerse: A cloud-based VR platform for research on multimodal human-robot interaction. *Front. Robot. AI* **2021**, *8*, 549360. [CrossRef] [PubMed]
38. ROS Sharp. 2022. Available online: https://github.com/siemens/ros-sharp (accessed on 10 October 2022).
39. ROS TCP Connector. 2022. Available online: https://github.com/Unity-Technologies/ROS-TCP-Connector (accessed on 10 October 2022).
40. Babaians, E.; Tamiz, M.; Sarfi, Y.; Mogoei, A.; Mehrabi, E. Ros2unity3d; high-performance plugin to interface ros with unity3d engine. In Proceedings of the 2018 9th Conference on Artificial Intelligence and Robotics and 2nd Asia-Pacific International Symposium, Kish Island, Iran, 10 December 2018; IEEE: Piscataway Township, NJ, USA, 2018; pp. 59–64.
41. Coronado, E.; Venture, G. Towards IoT-Aided Human–Robot Interaction Using NEP and ROS: A Platform-Independent, Accessible and Distributed Approach. *Sensors* **2020**, *20*, 1500. [CrossRef]
42. ZeroMQ Socket Api. 2022. Available online: https://zeromq.org/socket-api/ (accessed on 10 October 2022).

43. Photon Engine. 2022. Available online: https://www.photonengine.com/ (accessed on 10 October 2022).
44. Mirror Networking. 2022. Available online: https://mirror-networking.gitbook.io/docs/ (accessed on 10 October 2022).
45. Netcode for GameObjects. 2022. Available online: https://docs-multiplayer.unity3d.com/ (accessed on 10 October 2022).
46. Dimitropoulos, N.; Togias, T.; Michalos, G.; Makris, S. Operator support in human–robot collaborative environments using AI enhanced wearable devices. *Procedia Cirp* **2021**, *97*, 464–469. [CrossRef]
47. Togias, T.; Gkournelos, C.; Angelakis, P.; Michalos, G.; Makris, S. Virtual reality environment for industrial robot control and path design. *Procedia CIRP* **2021**, *100*, 133–138. [CrossRef]
48. Gao, Y.; Huang, C.M. PATI: A projection-based augmented table-top interface for robot programming. In Proceedings of the 24th International Conference on Intelligent User Interfaces, Marina del Ray, CA, USA, 17–20 March 2019; pp. 345–355.
49. TouchScript. 2022. Available online: https://github.com/TouchScript/TouchScript (accessed on 10 October 2022).
50. Aldoma, A.; Marton, Z.C.; Tombari, F.; Wohlkinger, W.; Potthast, C.; Zeisl, B.; Rusu, R.B.; Gedikli, S.; Vincze, M. Tutorial: Point cloud library: Three-dimensional object recognition and 6 dof pose estimation. *IEEE Robot. Autom. Mag.* **2012**, *19*, 80–91. [CrossRef]
51. Fischler, M.A.; Bolles, R.C. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM* **1981**, *24*, 381–395. [CrossRef]
52. Zhou, T.; Zhu, Q.; Du, J. Intuitive robot teleoperation for civil engineering operations with virtual reality and deep learning scene reconstruction. *Adv. Eng. Inform.* **2020**, *46*, 101170. [CrossRef]
53. PointNet. 2022. Available online: https://github.com/charlesq34/pointnet (accessed on 10 October 2022).
54. Vuforia Engine Package Unity. 2022. Available online: https://library.vuforia.com/getting-started/vuforia-engine-package-unity (accessed on 10 October 2022).
55. Chacko, S.M.; Granado, A.; Kapila, V. An augmented reality framework for robotic tool-path teaching. *Procedia CIRP* **2020**, *93*, 1218–1223. [CrossRef]
56. Google ARCore. 2022. Available online: https://developers.google.com/ar (accessed on 10 October 2022).
57. Mixed Reality Toolkit. 2022. Available online: https://github.com/microsoft/MixedRealityToolkit-Unity (accessed on 10 October 2022).
58. Lotsaris, K.; Gkournelos, C.; Fousekis, N.; Kousi, N.; Makris, S. AR based robot programming using teaching by demonstration techniques. *Procedia CIRP* **2021**, *97*, 459–463. [CrossRef]
59. Zxing. 2022. Available online: https://github.com/zxing/zxing (accessed on 10 October 2022).
60. Botev, J.; Rodríguez Lera, F.J. Immersive robotic telepresence for remote educational scenarios. *Sustainability* **2021**, *13*, 4717. [CrossRef]
61. IAI Kinect. 2022. Available online: https://nuitrack.com/ (accessed on 1 January 2023).
62. Bradski, G.; Kaehler, A. OpenCV. *Dr. Dobb's J. Softw. Tools* **2000**, *3*, 120.
63. Find Object 2D ROS package. 2022. Available online: http://wiki.ros.org/find_object_2d (accessed on 10 October 2022).
64. Moveit. *Motion Planning Framework*; 2020. Available online: https://moveit.ros.org/ (accessed on 10 October 2022).
65. IAI Kinect. 2022. Available online: https://github.com/code-iai/iai_kinect2 (accessed on 1 January 2023).
66. RobCog-IAI. 2022. Available online: https://github.com/robcog-iai (accessed on 1 January 2023).
67. Newton VR. 2022. Available online: https://assetstore.unity.com/packages/tools/newtonvr-75712 (accessed on 10 October 2022).
68. Li, R.; van Almkerk, M.; van Waveren, S.; Carter, E.; Leite, I. Comparing human-robot proxemics between virtual reality and the real world. In Proceedings of the 2019 14th ACM/IEEE international conference on human-robot interaction (HRI), Daegu, Korea, 11–14 March 2019; IEEE: Piscataway Township, NJ, USA, 2019; pp. 431–439.
69. Alonso, R.; Bonini, A.; Reforgiato Recupero, D.; Spano, L.D. Exploiting virtual reality and the robot operating system to remote-control a humanoid robot. *Multimed. Tools Appl.* **2022**, *81*, 15565–15592. [CrossRef]
70. Shariati, A.; Shahab, M.; Meghdari, A.; Amoozandeh Nobaveh, A.; Rafatnejad, R.; Mozafari, B. Virtual reality social robot platform: A case study on Arash social robot. In *Proceedings of the International Conference on Social Robotics*; Springer: Berlin/Heidelberg, Germany, 2018; pp. 551–560.
71. Pot, E.; Monceaux, J.; Gelin, R.; Maisonnier, B. Choregraphe: A graphical tool for humanoid robot programming. In Proceedings of the RO-MAN 2009-The 18th IEEE International Symposium on Robot and Human Interactive Communication, New Delhi, India, 14–18 October 2009; IEEE: Piscataway Township, NJ, USA, 2009; pp. 46–51.
72. Coronado, E.; Mastrogiovanni, F.; Indurkhya, B.; Venture, G. Visual programming environments for end-user development of intelligent and social robots, a systematic review. *J. Comput. Lang.* **2020**, *58*, 100970. [CrossRef]
73. Cao, Y.; Wang, T.; Qian, X.; Rao, P.S.; Wadhawan, M.; Huo, K.; Ramani, K. GhostAR: A time-space editor for embodied authoring of human-robot collaborative task with augmented reality. In Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology, New Orleans, LA, USA, 20–23 October 2019; pp. 521–534.
74. Ostanin, M.; Mikhel, S.; Evlampiev, A.; Skvortsova, V.; Klimchik, A. Human-robot interaction for robotic manipulator programming in Mixed Reality. In Proceedings of the 2020 IEEE International Conference on Robotics and Automation (ICRA), Paris, France, 31 May–31 August 2020; IEEE: Piscataway Township, NJ, USA, 2020; pp. 2805–2811.
75. Soares, I.; Petry, M.; Moreira, A.P. Programming Robots by Demonstration Using Augmented Reality. *Sensors* **2021**, *21*, 5976. [CrossRef]

76. Karan, M.S.; Berkman, M.İ.; Çatak, G. Smartphone as a Paired Game Input Device: An Application on HoloLens Head Mounted Augmented Reality System. In *Game+ Design Education*; Springer: Berlin/Heidelberg, Germany, 2021; pp. 265–277.
77. Mahmood, F.; Mahmood, E.; Dorfman, R.G.; Mitchell, J.; Mahmood, F.U.; Jones, S.B.; Matyal, R. Augmented reality and ultrasound education: Initial experience. *J. Cardiothorac. Vasc. Anesth.* **2018**, *32*, 1363–1367. [CrossRef]
78. Tian-Han, G.; Qiao-Yu, T.; Shuo, Z. The virtual museum based on HoloLens and vuforia. In Proceedings of the 2018 4th Annual International Conference on Network and Information Systems for Computers (ICNISC), Wuhan, China, 20–22 April 2018; IEEE: Piscataway Township, NJ, USA, 2018; pp. 382–386.
79. Solanes, J.E.; Muñoz, A.; Gracia, L.; Martí, A.; Girbés-Juan, V.; Tornero, J. Teleoperation of industrial robot manipulators based on augmented reality. *Int. J. Adv. Manuf. Technol.* **2020**, *111*, 1077–1097. [CrossRef]
80. Naceri, A.; Mazzanti, D.; Bimbo, J.; Tefera, Y.T.; Prattichizzo, D.; Caldwell, D.G.; Mattos, L.S.; Deshpande, N. The Vicarios Virtual Reality Interface for Remote Robotic Teleoperation. *J. Intell. Robot. Syst.* **2021**, *101*, 1–16. [CrossRef]
81. Su, Y.; Chen, X.; Zhou, T.; Pretty, C.; Chase, G. Mixed reality-integrated 3D/2D vision mapping for intuitive teleoperation of mobile manipulator. *Robot. Comput. Integr. Manuf.* **2022**, *77*, 102332. [CrossRef]
82. Whitney, D.; Rosen, E.; Ullman, D.; Phillips, E.; Tellex, S. Ros reality: A virtual reality framework using consumer-grade hardware for ros-enabled robots. In Proceedings of the 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Madrid, Spain; IEEE: Piscataway Township, NJ, USA, 2018; pp. 1–9.
83. Palmarini, R.; del Amo, I.F.; Bertolino, G.; Dini, G.; Erkoyuncu, J.A.; Roy, R.; Farnsworth, M. Designing an AR interface to improve trust in Human-Robots collaboration. *Procedia CIRP* **2018**, *70*, 350–355. [CrossRef]
84. Wang, X.V.; Wang, L.; Lei, M.; Zhao, Y. Closed-loop augmented reality towards accurate human-robot collaboration. *CIRP Ann.* **2020**, *69*, 425–428. [CrossRef]
85. Mahadevan, K.; Sousa, M.; Tang, A.; Grossman, T. "grip-that-there": An investigation of explicit and implicit task allocation techniques for human-robot collaboration. In Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems, Yokohama, Japan, 8–13 May 2021; pp. 1–14.
86. Chandan, K.; Kudalkar, V.; Li, X.; Zhang, S. ARROCH: Augmented reality for robots collaborating with a human. In Proceedings of the 2021 IEEE International Conference on Robotics and Automation (ICRA), Xi'an, China, 30 May–5 June 2021; IEEE: Piscataway Township, NJ, USA, 2021; pp. 3787–3793.
87. Weber, D.; Kasneci, E.; Zell, A. Exploiting Augmented Reality for Extrinsic Robot Calibration and Eye-based Human-Robot Collaboration. In Proceedings of the 2022 ACM/IEEE International Conference on Human-Robot Interaction, Sapporo, Hokkaido, Japan, 7–10 March 2022; pp. 284–293.
88. Liu, S.; Wang, X.V.; Wang, L. Digital twin-enabled advance execution for human-robot collaborative assembly. *CIRP Ann.* **2022**, *71*, 25–28 [CrossRef]
89. Tuli, T.B.; Manns, M.; Zeller, S. Human motion quality and accuracy measuring method for human–robot physical interactions. *Intell. Serv. Robot.* **2022**, 15, 1–10. [CrossRef]
90. Aivaliotis, S.; Lotsaris, K.; Gkournelos, C.; Fourtakas, N.; Koukas, S.; Kousi, N.; Makris, S. An augmented reality software suite enabling seamless human robot interaction. *Int. J. Comput. Integr. Manuf.* **2022**, 35, 1–27. [CrossRef]
91. Lotsaris, K.; Fousekis, N.; Koukas, S.; Aivaliotis, S.; Kousi, N.; Michalos, G.; Makris, S. Augmented reality (ar) based framework for supporting human workers in flexible manufacturing. *Procedia CIRP* **2021**, *96*, 301–306. [CrossRef]
92. Malik, A.A.; Brem, A. Digital twins for collaborative robots: A case study in human-robot interaction. *Robot. Comput. Integr. Manuf.* **2021**, *68*, 102092. [CrossRef]
93. Wang, L.; Liu, S.; Cooper, C.; Wang, X.V.; Gao, R.X. Function block-based human-robot collaborative assembly driven by brainwaves. *CIRP Ann.* **2021**, *70*, 5–8. [CrossRef]
94. Rebenitsch, L.; Owen, C. Estimating cybersickness from virtual reality applications. *Virtual Real.* **2021**, *25*, 165–174. [CrossRef]
95. Vosniakos, G.C.; Ouillon, L.; Matsas, E. Exploration of two safety strategies in human-robot collaborative manufacturing using Virtual Reality. *Procedia Manuf.* **2019**, *38*, 524–531. [CrossRef]
96. Montalvo, W.; Bonilla-Vasconez, P.; Altamirano, S.; Garcia, C.A.; Garcia, M.V. Industrial Control Robot Based on Augmented Reality and IoT Protocol. In Proceedings of the International Conference on Augmented Reality, Virtual Reality and Computer Graphics, Virtual Event, 7–10 September 2020; Springer: Berlin/Heidelberg, Germany, 2020; pp. 345–363.
97. Botev, J.; Rodríguez Lera, F.J. Immersive Telepresence Framework for Remote Educational Scenarios. In *Proceedings of the International Conference on Human-Computer Interaction*; Springer: Berlin/Heidelberg, Germany, 2020; pp. 373–390.