frontiers | Frontiers in Virtual Reality

TYPE Original Research
PUBLISHED 04 December 2023
DOI 10.3389/frvir.2023.1210211

OPEN ACCESS

EDITED BY
Dioselin Gonzalez,
Independent Researcher, United States

REVIEWED BY
Federico Manuri,
Polytechnic University of Turin, Italy
Bill Pike,
United States Army, United States

*CORRESPONDENCE
Silvia Ferrari,
ferrari@cornell.edu

†These authors have contributed equally to this work and share first authorship

RECEIVED 21 April 2023
ACCEPTED 07 July 2023
PUBLISHED 04 December 2023

CITATION
Paradise A, Surve S, Menezes JC, Gupta M, Bisht V, Jang KR, Liu C, Qiu S, Dong J, Shin J and Ferrari S (2023), RealTHASC—a cyber-physical XR testbed for AI-supported real-time human autonomous systems collaborations. *Front. Virtual Real.* 4:1210211. doi: 10.3389/frvir.2023.1210211

RealTHASC—a cyber-physical XR testbed for AI-supported real-time human autonomous systems collaborations

Andre Paradise[1†], Sushrut Surve[1†], Jovan C. Menezes[1], Madhav Gupta[1], Vaibhav Bisht[1], Kyung Rak Jang[1], Cong Liu[1], Suming Qiu[1], Junyi Dong[1], Jane Shin[2] and Silvia Ferrari[1*]

[1]Laboratory for Intelligent Systems and Controls (LISC), Sibley School of Mechanical and Aerospace Engineering, Cornell University, Ithaca, NY, United States, [2]Active Perception and Robot Intelligence Lab (APRILab), Department of Mechanical and Aerospace Engineering, University of Florida, Gainesville, FL, United States

Today's research on human-robot teaming requires the ability to test artificial intelligence (AI) algorithms for perception and decision-making in complex real-world environments. Field experiments, also referred to as experiments "in the wild," do not provide the level of detailed ground truth necessary for thorough performance comparisons and validation. Experiments on pre-recorded real-world data sets are also significantly limited in their usefulness because they do not allow researchers to test the effectiveness of active robot perception and control or decision strategies in the loop. Additionally, research on large human-robot teams requires tests and experiments that are too costly even for the industry and may result in considerable time losses when experiments go awry. The novel Real-Time Human Autonomous Systems Collaborations (RealTHASC) facility at Cornell University interfaces real and virtual robots and humans with photorealistic simulated environments by implementing new concepts for the seamless integration of wearable sensors, motion capture, physics-based simulations, robot hardware and virtual reality (VR). The result is an extended reality (XR) testbed by which real robots and humans in the laboratory are able to experience virtual worlds, inclusive of virtual agents, through real-time visual feedback and interaction. VR body tracking by DeepMotion is employed in conjunction with the OptiTrack motion capture system to transfer every human subject and robot in the real physical laboratory space into a synthetic virtual environment, thereby constructing corresponding human/robot avatars that not only mimic the behaviors of the real agents but also experience the virtual world through virtual sensors and transmit the sensor data back to the real human/robot agent, all in real time. New cross-domain synthetic environments are created in RealTHASC using Unreal Engine™, bridging the simulation-to-reality gap and allowing for the inclusion of underwater/ground/aerial autonomous vehicles, each equipped with a multi-modal sensor suite. The experimental capabilities offered by RealTHASC are demonstrated through three case studies showcasing mixed real/virtual human/robot interactions in diverse domains, leveraging and complementing the benefits of experimentation in simulation and in the real world.

KEYWORDS

robotics, virtual reality, human-autonomy teams, simulation systems, human-robot interaction, multi-robot communication, simulation-to-reality gap, artificial intelligence

1 Introduction

Emerging techniques in machine learning (ML), AI, and computer vision are able to equip the next-generation of autonomous robots with unprecedented sensing, cognitive, and decision-making skills (Bassyouni and Elhajj, 2021). These "smart" robots will inevitably interact with humans as part of collaborative robot teams in unstructured and dynamic environments, thus requiring fundamental cross-cutting research not only in engineering and computer science but also in the humanities (Fong et al., 2003; Pendleton et al., 2017; Liu et al., 2021). Real-world tests with human subjects in the loop are necessary in almost every sector of robotics, including defense, agriculture, healthcare, and emergency-response systems, to name a few. Humans that collaborate with AI software and/or autonomous robot teammates, also known as human-autonomy teams (HATs), will soon be required to solve complex and safety-critical tasks (Oh et al., 2017) such as target localization and mapping (Krajník et al., 2014), fire fighting (Naghsh et al., 2008), search and rescue (SpurnỲ et al., 2019), security and surveillance (Hu and Lanzon, 2018), and cooperative transportation (Chen et al., 2015). To date, controlled simulation and laboratory experiments have been primarily used by researchers in both industry and academia in order to develop and test new theories and algorithms. As a result, when introduced into real-world applications with humans-in-the-loop, they all too often fail because of unanticipated difficulties and environments. Deploying new decision and control algorithms in the field is not only costly but also poses safety hazards when humans are present and, at best, fails to provide researchers with the knowledge, control, and post-processing capabilities required during the development phase. Recent advancements in data-driven approaches for robotics applications also require large amounts of training data that may not always be available from real-world sensors (Choi et al., 2021).

In response to the aforementioned difficulties, various computer simulation platforms have emerged in the field of robotics and HAT collaborations (Puig et al., 2018; Deitke et al., 2020; Shen et al., 2021). These platforms have played a crucial role in the iterative development of autonomous robots by providing a safe and controlled environment for testing, training, and refining AI algorithms and behaviors. Simulation environments have also allowed researchers and developers to explore a wide range of scenarios, manipulate variables, and assess the performance of robot control and decision algorithms in a cost-effective and time-efficient manner. By testing physical and cognitive functions such as sensing, perception and control, simulation systems have enabled the evaluation and comparison of different robot design choices and interaction strategies. Virtual replicas of real-world robots, also known as digital twins, have in fact become increasingly popular in both industry and academia for research on robot control and operation (Garg et al., 2021). Traditional robot simulators such as Webots (Michel, 2004) and Gazebo (Koenig and Howard, 2004) rely on physics-based Open Dynamics Engine (ODE) for replicating robot motion and on 3D rendering engine for constructing the robot environment (Erez et al., 2015). Although this architecture has proven extremely useful to date, there remains a significant simulation-to-reality gap for testing more complex systems of human-robot teams (Škulj et al., 2021). In order to aid in the development of robust autonomy algorithms, simulation environments must address major challenges, including transferring knowledge from the virtual world to the real world; developing stochastic and realistic scenes; and developing fully interactive, multi-AI agent, scalable 3D environments (Reiners et al., 2021). Off-the-shelf game development software, such as Unity and Unreal Engine™ 4 (UE™) (Epic Games, 2019), has been gaining increasing attention because of its photorealistic rendering capabilities. New simulation frameworks such as AirSim (Shah et al., 2018), UnrealCV (Qiu and Yuille, 2016), and CARLA (Dosovitskiy et al., 2017) have also demonstrated success in training and verifying computer vision and perception-based algorithms. Additionally, leveraging VR technologies alongside such platforms has enabled studies on collaboration between humans and robots mimicked by digital twins (Mizuchi and Inamura, 2017; Inamura and Mizuchi, 2021; Murnane et al., 2021).

In particular, FlightGoggles, a Unity-based photorealistic sensor simulator for perception-driven robots, has begun to bridge the simulation-to-reality gap (Guerra et al., 2019). This previous work developed a novel hardware-in-the-loop approach by placing a real vehicle and human in a motion capture facility and augmenting them with an obstacle-populated virtual environment. This framework successfully created a photorealistic virtual world in which both robot and human perception occurred virtually. Additionally, FlightGoggles successfully implemented multiple simulated robots to demonstrate their computational capabilities within a novel system architecture. As autonomous robots and intelligent machines are becoming an integral part of collaborative human teams, a significant body of research has focused on identifying effective means of human-robot communication, task allocation, and multi-robot coordination (Gemerek et al., 2019; Ognibene et al., 2022). While FlightGoggles has proven the capability to incorporate robots and humans into a desired computing architecture, it does not yet support complex perception and control human-robot interactions in mixed real/virtual worlds.

The RealTHASC facility presented in this paper provides a new extended reality testbed within which researchers may investigate complex real-time human-robot and multi-robot interactions based on tailored combinations of exteroceptive and proprioceptive sensors installed on virtual and real-world agents. Its main contribution is a novel framework for integrating VR technology, embedded systems design, and computer graphics tools to enable safe, real-time, and photorealistic multi-agent interaction for collaborative decision-making in HATs. Novel software and hardware integration tools are required to support real-time perception feedback from virtual avatars to real agents, with information-driven planning and control in the loop. The ability to support perception-based control, real-time communication between real and virtual agents, and HAT interactions and collaboration in challenging test environments is also demonstrated. The formulation and nomenclature necessary for describing the real and virtual workspace, along with their respective agents, are introduced in Section 2. The architecture of the testbed, including the simulation environment, human interface, communication between the agents, and robot perception and planning, is described in Section 3. To demonstrate the modality of RealTHASC, three applications encompassing human-robot and multi-robot interaction are presented and analyzed in Section 4. Finally, possible future research directions are discussed in Section 5.

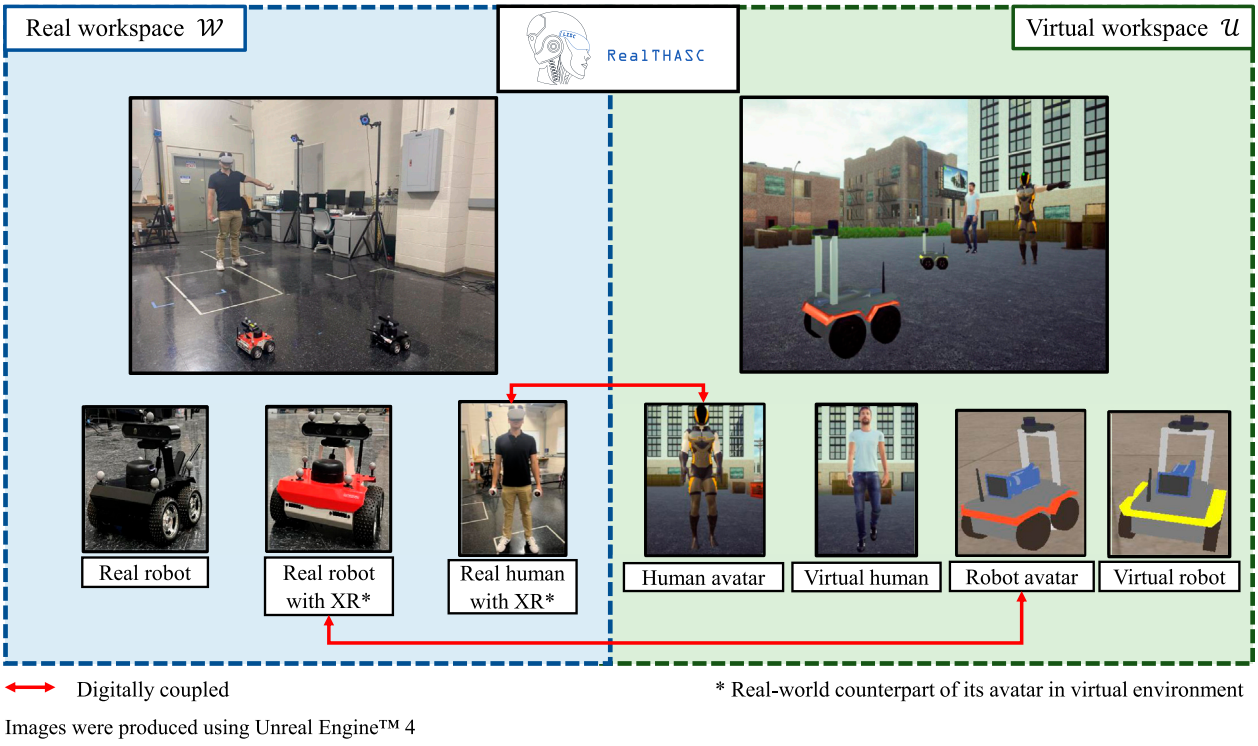


FIGURE 1
Real and virtual agents in the RealTHASC facility, where real agents with XR are digitally coupled with virtual avatars in Unreal Engine™.

2 Mathematical preliminaries and notation

The RealTHASC facility consists of a novel HAT testbed comprised of collaborative agents, operating and interacting in and across physical and simulated worlds. Let the experimental region located inside the physical laboratory, defined as the real workspace, be denoted by $\mathcal{W} \subset \mathbb{R}^3$ and let the simulated environment created in Unreal Engine™, defined as the virtual workspace, be denoted by $\mathcal{U} \subset \mathbb{R}^3$. Based on the desired type of HAT-environment interactions, four types of agents can be introduced into RealTHASC: real agents, virtual agents, avatars, and real agents with XR. *Real human/robot agents* operate and sense solely in $\mathcal{W}$, whereas *virtual human/robot agents* operate and sense solely in $\mathcal{U}$. *Avatars* sense in $\mathcal{U}$ and transmit their sensor data back to their corresponding *real agent with XR*, based on the avatar's position and orientation in $\mathcal{U}$. Importantly, real agents with XR are kinematically coupled with their avatars. Hence, not only their sensors have the same field-of-view (FOV) as those of their avatars, but also they can interact with elements and agents in the virtual workspace through the avatar's behavior, react to the perception feedback, and the full body resulting state is relayed to their avatar in real time. Examples of RealTHASC agents operating in mixed environments that are comprised of real and virtual workspaces are shown in Figure 1.

Let R and H denote the index sets of robots and humans operating in $\mathcal{W}$, where each robot and human is associated with an index $i \in R$ and $j \in H$, respectively. Also, let P and Q denote the index sets of robots and humans operating in $\mathcal{U}$, where each robot and human is associated with an index $i \in P$ and $j \in Q$, respectively. Real agents with XR are denoted by the same indices in R and H as their avatars in P and Q, respectively. The homogenous rigid-body transformation which maps every point of $\mathcal{W}$ into $\mathcal{U}$ is denoted by $\mathbf{T}$, where its inverse $\mathbf{T}^{-1}$ is assumed to exist and both $\mathbf{T}$ and $\mathbf{T}^{-1}$ are known *a priori* from the RealTHASC setup. In this paper, unmanned ground vehicles (UGVs) are used as robot agents to showcase the capabilities of RealTHASC. The framework, however, can be easily extended to any other agents including autonomous underwater vehicles (AUVs) and autonomous aerial vehicles (AAVs). In demonstrations presented in this paper, human and robot agents all move on the ground plane, which is coplanar to the XY plane of the inertial frames $\mathcal{F}_{\mathcal{W}}$ and $\mathcal{F}_{\mathcal{U}}$, embedded in $\mathcal{W}$ and $\mathcal{U}$ respectively. The configuration of robot $i \in \{R \cup P\}$ is denoted by $\mathbf{q}_i = [x_i \quad y_i \quad \theta_i]^T$, and the pose of human j, comprised of his/her position and heading, is denoted by $\mathbf{s}_j = [x_j \quad y_j \quad \theta_j]^T$. The configuration of all agents operating in $\mathcal{W}$ is estimated using a motion capture system or VR tracking, whereas the configurations of agents in $\mathcal{U}$ are known without error from the UE™ simulation environment. In the demonstrations presented in this paper, every robot $i \in \{R \cup P\}$ obeys the unicycle motion model,

$$\dot{\mathbf{q}}_i = \begin{bmatrix} \dot{x}_i \\ \dot{y}_i \\ \dot{\theta}_i \end{bmatrix} = \begin{bmatrix} v_i \cos\theta_i \\ v_i \sin\theta_i \\ \omega_i \end{bmatrix} = \mathbf{f}(\mathbf{q}_i, \mathbf{u}_i), \tag{1}$$

where the control vector consists of the robot linear velocity v_i and angular velocity ω_i, or $\mathbf{u}_i = [v_i \quad \omega_i]^T$. All the aforementioned notation is summarized in Figure 2.

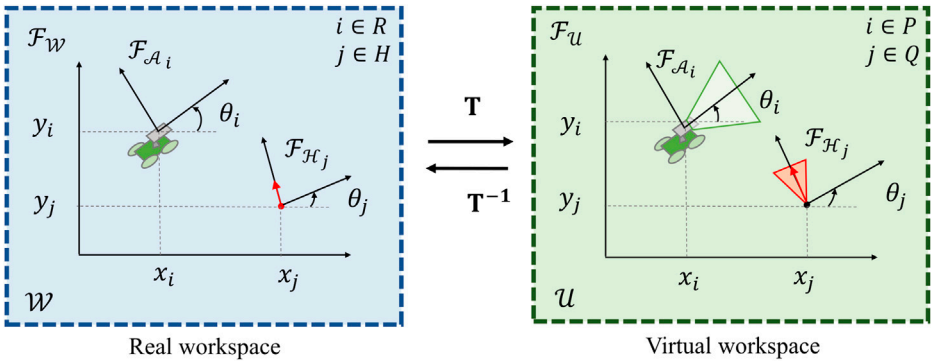


FIGURE 2
Configurations of a human and robot agent in the real workspace (left) and their corresponding avatars in the virtual workspace (right).

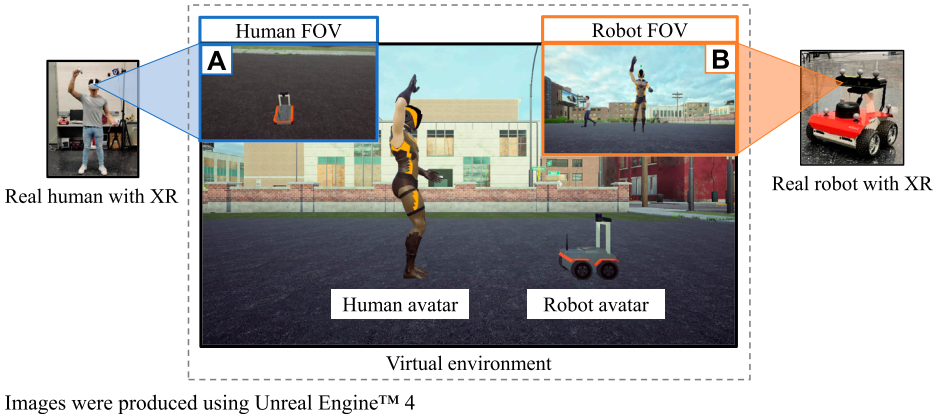


FIGURE 3
Visual perception of industrial city virtual environment in which the avatars of **(A)** human and **(B)** robot agents interact in real time.

3 System architecture

Within RealTHASC, physical robots and real humans interact and communicate with simulated agents as well as avatars in photorealistic virtual environments (Section 3.1). A key capability of the facility is that both simulated agents and avatars sense the virtual environments by means of UE™ synthetic sensors, as shown in Figure 3, while possibly operating in potentially different real workspaces. Integrative cyber-physical interfaces (Sections 3.2–3.4) are leveraged to achieve this capability, thus enabling collaborative decision-making across real and synthetic worlds. This novel integration allows virtual environments to act as a common medium for safe yet realistic real-time inter-agent and agent-environment interactions. These interactions can then be modeled, analyzed, and leveraged for various applications without any restrictions imposed by the considerations normally associated with testing robots in the lab or in the wild, including safety and reproducibility. RealTHASC uses the 3D graphics development software, UE™, to create virtual environments. This provides the flexibility to test algorithms and collect data in a wide variety of environments such as subways, cities, offices, and oceans under varying lighting and weather conditions along with a diverse set of user-defined static and dynamic obstacles. The RealTHASC framework supports multiple autonomous and user-controlled agents, enabling complex online multi-agent control and coordination experiments. Example demonstrations of mixed policies implemented for the planning and control of both virtual and real robots are described in Section 3.5.

Figure 4 shows the RealTHASC framework used to achieve real-time collaboration between real robots and humans via their avatars in a virtual environment developed in UE™. The blue arrows indicate how the kinematics of the real-world agents are communicated to their respective avatars while the orange arrows show how the perception feedback of these avatars is communicated to their real-world counterparts. Human operators in $\mathcal{W}$ control their human avatars in $\mathcal{U}$ using real-time VR body tracking enabled by a VR headset and handheld controllers. The robots operating in $\mathcal{W}$ are equipped with reflective markers detected by the motion capture system installed in this workspace, which streams the real robot state to their respective robot avatars. Virtual sensors, defined using sensor application programming interfaces (APIs) as

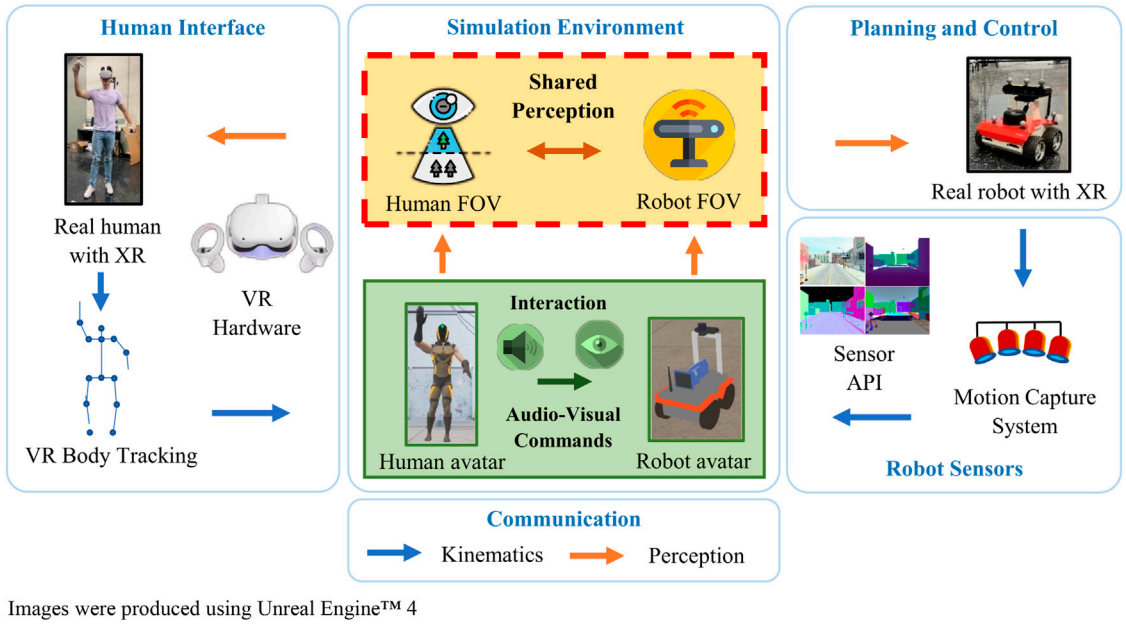


FIGURE 4
The RealTHASC framework enables research on human-autonomy interactions and collaborations by importing both human and robot avatars, with sensor suites, into complex and photorealistic simulation environments with perfect ground truth.

TABLE 1 Description of the hardware and software components used to build the RealTHASC facility.

Component	Application
Hardware	
10x OptiTrack Prime^x 22 Camera	Multi-camera network localizes and tracks physical robots using placed markers
2x ROSbot 2.0	UGV for the physical robot experiments
Meta Quest 2	VR platform allows human users to see, hear, and interact with virtual worlds
Dell Alienware Aurora R13	Primary desktop computer acts as the head substation hosting the simulation environment and connections from the laboratory environment
2x Alienware x15 R2 Gaming Laptop	Base control station laptop that receives waypoint (x, y, θ) from the desktop and communicates control command to the ROSbot
Software	
Unreal Engine™ 4.24	3D computer graphics game engine hosts the virtual environments alongside virtual human and robot actors
Motive 3.0	Skeletal solver creates rigid bodies for robots using tracked markers
DeepMotion SDK	Three-point VR tracker transfers human movement to UE4™ actor
NatNet SDK	Transfers localization information from Motive to UE4™ and ROSbot

discussed in Section 3.3, are used to simulate the robot FOV while the humans observe the workspace through the VR headset. Since the human and robot FOVs are defined inside the virtual environment, they can be constantly monitored in real time to facilitate shared perception in agents. This enables direct sharing of visual cues observed by any agent with any other agent, human or robot, in contrast to existing facilities which only consider sharing cues inside the robot FOV (Nourbakhsh et al., 2005). Additionally, humans in the RealTHASC facility can provide audio or visual commands to real and virtual robots in support of research and testing on human-robot interactions, as explained in Section 4.1. A summary of all hardware and software components used to create the RealTHASC facility is shown in Table 1.

3.1 RealTHASC UE™ simulation environment

In the RealTHASC facility, the UE™ simulation environment acts as the interface between the real and virtual agents. The UE™ software is chosen because it is currently considered as the most visually realistic tool for bridging the simulation-to-reality gap in perception-related tasks. The framework shown in Figure 1

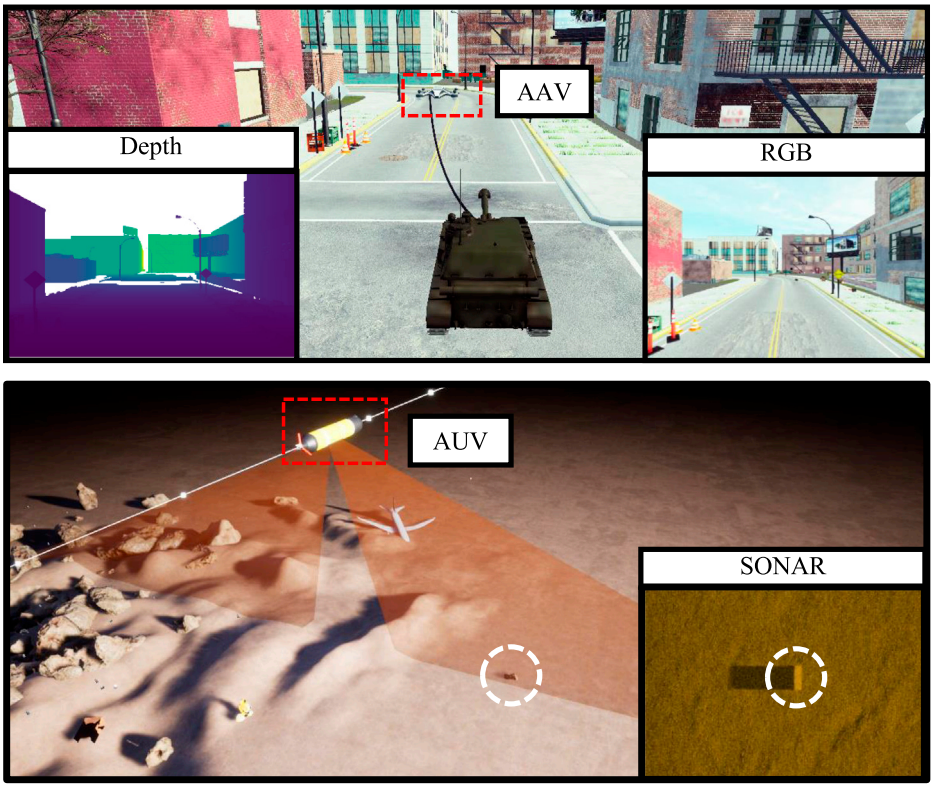


FIGURE 5
Virtual worlds, inclusive of autonomous robots, sensors, and artificial intelligence algorithms, are developed exclusively for RealTHASC using UE4™. Examples of these synthetic environments include **(A)** an industrial city monitored by an AAV equipped with onboard RGBD camera and **(B)** an underwater environment scanned by an AUV equipped with a side-scan sonar. © 2022 IEEE. Reprinted, with permission, from Shin et al. (2022a).

leverages UE™ to support real-time rendering and manipulation of multiple programmable, photorealistic environments. The base environments used for the experiments presented in Section 4, namely, the industrial city (Figure 5A) and the undersea environment (Figure 5B), are obtained from the UE™ Marketplace. These base environments are modified to include virtual agents and digital avatars, to support their associated sensing modalities, and to stream data amongst various agents of the HATs (Section 3.4). The ability to programmatically manipulate environmental conditions such as fog, time of day, and luminosity, while difficult in real-world or laboratory physical experiments, is easily accomplished within RealTHASC. As a result, real humans and robots can interact with and test a broad range of environmental conditions known to influence visual perception and active control tasks.

Digital avatars are created to resemble their real-world counterparts aesthetically, as required by the chosen experimental test or scenario, and also functionally by establishing a kinematic and sensing coupling with their real-world counterparts, as explained in Section 3.2 and Section 3.3. C++ programmable actors such as virtual robots, pedestrians, and mobile vehicles can be controlled offline using predefined trajectories or are equipped with simulated dynamics, perception, and control algorithms running online to test autonomy and collaboration algorithms with hardware or software-in-the-loop, as explained in Section 3.3 and Section 3.5 respectively. Two examples of such programmed actors with feedback controller-in-the-loop, an AAV and AUV, are shown in Figure 5. The AAV and AUV actors both sense their environment by the means of onboard RGBD camera and SONAR, respectively.

3.2 RealTHASC human interface

RealTHASC allows human operators to perceive and interact with synthetic environments and autonomous agents therein by synchronizing body movements with their avatars in real time. For instance, humans may need to react to robot motions and behaviors while also providing commands to their robot teammates in the HAT by means of semantics or hand gestures. Additionally, real robots and humans are able to interact with virtual humans and other human avatars in the UE™ world, allowing one to test collaborative tasks performed by larger HATs that may include real humans and robots at different geographic locations. Human avatars are simulated using the Meta Quest 2 hardware with a Steam VR backend. Using the VR headset, a real human (with XR) is able to view the rendered frames from the simulation environment and listen to audio, as sensed by the human avatar. This integration provides the user with an immersive, interactive first-person experience of the simulation environment. The headset and accompanying handheld controllers have trackers which

communicate their positions over Wi-Fi to the system running Steam VR connected with UE™.

The DeepMotion SDK (DeepMotion, 2023) is used to transform this data into joint motions of a predefined skeleton of a human avatar, using three-point VR body tracking. By this approach, human operators inside the RealTHASC facility are able to control their virtual avatars without attaching any extra markers to the body with a mean latency of 10 ms. This facilitates real-time seamless kinematic coupling and interaction. The appearance of human avatars becomes crucial when testing algorithms trained on real-world data sets obtained from application-driven environments (e.g., offices and industrial workshops) or when using UE™-based worlds and simulations to produce synthetic datasets. In order to solve the problem of simulation-to-reality and reality-to-simulation transfer, a user interface is built using the UE™ Blueprint to easily modify and select between avatars of interest depending on the chosen domain or HAT application.

3.3 RealTHASC robot sensing

One of the main goals of RealTHASC is to enable research on active sensor systems used to gather, process, and communicate information about their operating environments to their encompassing and surrounding agents (Ferrari and Wettergren, 2021). The broad range of sensing modalities available on real and virtual robots play a pivotal role in allowing for many types of HAT collaborations to be synthesized inside the facility. RealTHASC hosts a unique array of proprioceptive sensors, measuring the ego state of the robot, and exteroceptive sensors, measuring the state of the operating environment. Husarion ROSbots, powered by ARM processors, are used as the UGV robot agents operating in $\mathcal{W}$. These real robots are equipped to localize using dead reckoning (Kao, 1991) enabled through rotary encoders and inertial measurement units (IMUs). These sensors are subject to drift and may be used to simulate navigation in GPS-denied scenarios. This facility has also been equipped with the OptiTrack motion capture system which provides robot localization within 10 mm accuracy using the reflective markers mounted on the robots. As shown in Figure 7, this information is streamed to the robot base control stations in real time, which allows the robots to use either the motion capture system or dead reckoning for localization. Since the real lab's workspace includes multiple robots, obstacles, and humans, the FOV of the motion capture cameras observing reflective markers can sometimes be blocked leading to loss of localization. In order to account for such cases, a localization strategy is designed that autonomously switches to dead-reckoning localization initialized at the last known localization of the robot and back to motion capture-based localization when the cameras see the markers.

Various modalities for perception are developed by integrating sensor APIs from UnrealCV (Qiu and Yuille, 2016), traditional computer vision algorithms, and recent advances in sensor modeling. This integration enables the simulation of three-channel 8-bit data streams including RGB cameras, online panoptic segmentation, and surface normal estimation. One-channel (16-bit) images of ground truth depth are also acquired in real time using predefined depth cameras and stereo RGB cameras in UnrealCV. An online processing pipeline is defined to process RGB data streams from UnrealCV to generate 16-bit (two 8-bit channels) images of dense optical flow using the Farneback estimation algorithm and 8-bit (one-channel) grayscale images using OpenCV (Bradski, 2000). This work also incorporates the image-based side scan sonar simulation by Shin et al. (2022a) to simulate underwater sensing in these virtual environments created in UE™. This simulation incorporates visual approximations for acoustic effects such as back-scatter and acoustic shadow to provide a realistic rendering of sonar sensors. Novel modular C++ programmable robot agents are created in UE™ by interfacing the aforementioned sensors with robot avatars and virtual robots to enable robot perception in the simulation environment. A separate Python script has been created to enable these sensors to be used as static sensors monitoring the environment or to be programmed to move on predefined trajectories for data collection. Illustrative examples for the type of sensing modalities available in the simulation environment are shown in Figure 6. All the real robots (without avatars) sense the real workspace and are equipped with RP LIDAR A2 (laser range scanner), Orbecc Astra RGBD camera, and Time-of-Flight (TOF) sensors. The output of these sensors mounted on the real and virtual robots is communicated over ROS to robot planners (Section 3.5) running onboard for active perception tasks facilitating inter-agent interaction.

3.4 Communication between the real and virtual workspace

The RealTHASC testbed hosts communication channels for message passing to facilitate inter-agent communication in and across the real and virtual workspace. An overview of the message-passing framework is described in Figure 7. An Alienware Aurora R13 system acts as the head substation that harbors communications from the OptiTrack motion capture cameras, hosts the virtual environment in UE™, and supports the VR tracking and associated hardware. As a result of the interactions between the agents in the virtual environment, the head substation generates desired waypoints for the robots based on the cooperation strategy employed. The waypoints intended for the robots in $\mathcal{W}$ are communicated to their respective base control stations over a local area network (LAN). The planning and control framework running on these stations, described in Section 3.5, converts these waypoints to control commands transmitted to the real robots over 2.4 GHz Wi-Fi networks. These Wi-Fi networks are set up on distinct custom channels to avoid aliasing from simultaneously sent control commands.

The base control stations communicate with each other over a LAN connection using User Datagram Protocols (UDP) to simulate inter-robot communication in real time with low latency. Since the robots move in the real workspace, the OptiTrack motion capture cameras stream data packets comprising the marker data to the head substation through a bridge at a frequency of 120 Hz. Motive, a proprietary OptiTrack data processing software, then uses this marker data to infer the localization of user-specified rigid bodies defined as a collection of markers. This inferred robot localization is streamed over UDP channels to real robots through their base control stations for planning purposes and to the robot avatars in $\mathcal{U}$ for kinematically coupling them with their real-world counterparts. This kinematic coupling between real robots with XR and their robot avatars is accomplished in real time using the proprietary NatNet SDK to stream rigid body localization data at 120 Hz with a latency of 10 m. The overall system time delay from

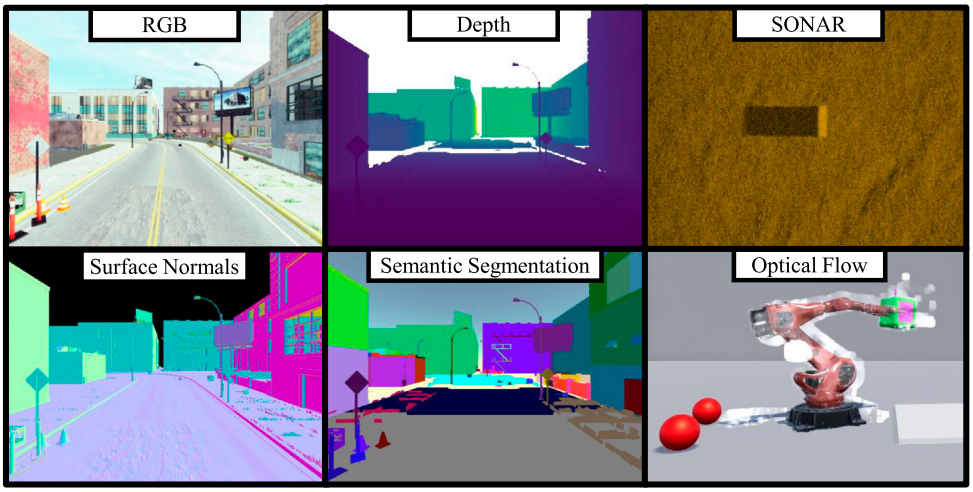


FIGURE 6
Examples of simulated sensing modalities and computer vision algorithms in RealTHASC. © 2022 IEEE. Reprinted, with permission, from Shin et al. (2022a).

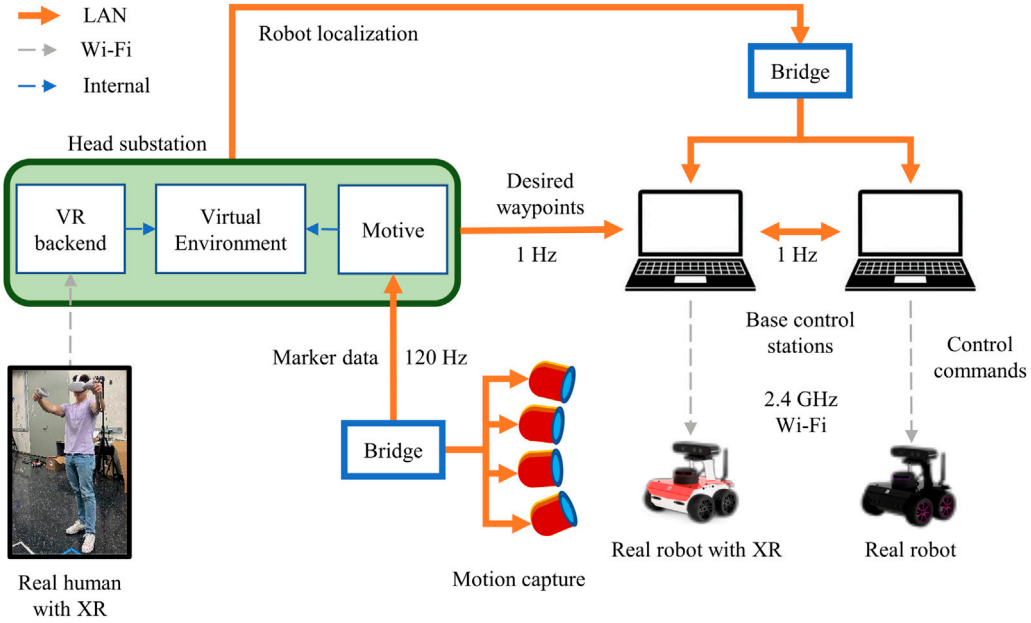


FIGURE 7
RealTHASC cyber-physical interfaces and communication pipelines enable seamless integration of real agents and applicable avatars in the virtual workspace.

sending a desired waypoint to detect the corresponding effects in the robot state is approximately 20 m.

3.5 Real-time robot planning and control

The ability to readily test and simulate collaborative planning and control algorithms for HAT research is demonstrated by implementing a waypoint-following policy on real and virtual robots in this testbed. For a robot $i \in R$ operating in $\mathcal{W}$ (real robots and real robots with XR), its base control station receives a waypoint denoted by $\mathbf{q}_i^*$, $i \in R$ in $\mathcal{W}$ directly or receives a waypoint in $\mathcal{U}$ and maps it to a desired waypoint in $\mathcal{W}$ using the transformation $\mathbf{T}^{-1}$. Subsequently, the robot executes the waypoint-following policy to determine appropriate control commands based on the received waypoint information. For virtual robots, however, the waypoint $\mathbf{q}_i^*$, $i \in P$ is directly used by the policy. For brevity, agent indices are omitted in the remainder of this section. Assuming $\mathbf{q}^*(k) =$

$[\mathbf{p}^* \quad \theta^*]^T$ is the desired waypoint for a robot in $\mathcal{W}$ at time-step k, the desired position and desired orientation is then $\mathbf{p}^* = [x^* \quad y^*]^T$ and θ^*, respectively. The chosen algorithm for waypoint-following implementation is based on a move-then-turn policy for each robot in the real and virtual workspace. At state $\mathbf{q} = [x \quad y \quad \theta]^T$, the robot first turns to point towards the desired waypoint position $\mathbf{p}^*$, moves towards it, and then rotates to reach the desired orientation θ^*. This policy outputs the control command $\mathbf{u} = [v \quad \omega]^T$ comprised of linear velocity v and angular velocity ω, which has been summarized in the order of execution as follows

$$v = 0 \qquad \omega = k_\theta \left(\tan^{-1}\left(\frac{y^* - y}{x^* - x}\right) - \theta \right) \quad (2)$$

$$v = k_x \ (x^* - x) + k_y \ (y^* - y) \qquad \omega = 0 \quad (3)$$

$$v = 0 \qquad \omega = k_\theta \ (\theta^* - \theta) \quad (4)$$

where k_x, k_y, $k_\theta \in \mathbb{R}^+$ are user-defined parameters for which larger values represent faster response to errors in robot pose. Simultaneously, the motion capture system continuously tracks the motion of these robots in $\mathcal{W}$ and records their states. As mentioned in Section 3.4, this localization information $\hat{\mathbf{q}}_i$, $i \in R$, an estimate of $\mathbf{q}_i$, is then streamed to the robot $i \in R$, via its base control station, and to its robot avatar $i \in P$, in UE™. The robot avatar is then moved to $\hat{\mathbf{q}}_i$, $i \in P$, which is calculated using the transformation $\mathbf{T}$ as shown in Figure 4. This control loop used to couple the real robot with its avatar runs at a frequency of 120 Hz, in real time. The planner continuously streams waypoints with the desired run-rate frequency for various robot tasks, explored further in Section 4, while the control policy outputs the appropriate command for the most recent waypoint.

4 Experiments

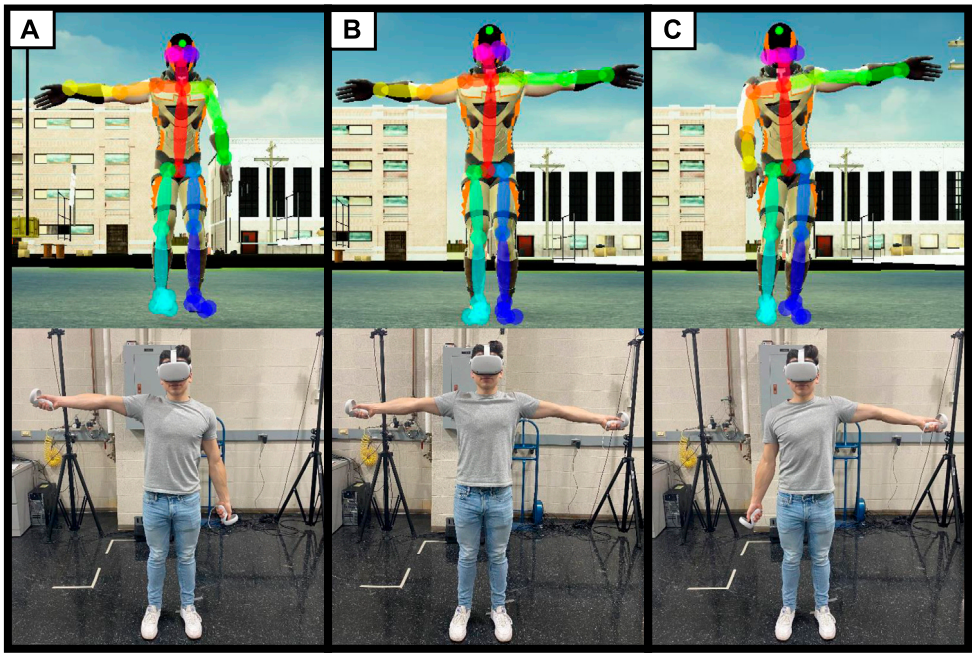
To convey the functionalities and capabilities of the RealTHASC facility, three experiments are conducted that each highlight different types of agent interactions across the real and virtual workspace. The first experiment focuses on human-robot interaction, the second experiment tests multi-robot teaming in and across real and virtual workspaces, and the third experiment showcases human-robot collaboration using synthetic sonar sensor simulation.

4.1 Human-robot perception and control

This experiment is designed to showcase interaction-based control of virtual robots and robot avatars using gesture commands from a human teammate. This demonstration takes place in the industrial city environment, built in UE™, hosting the following actors: a human avatar, a robot avatar, and a virtual robot. As shown in Figure 3, all agents perceive the virtual workspace using simulated RGB cameras. Human avatars communicate with the robot agents using gestures, as shown in Figure 8, to command the next waypoint. These gestures are detected by a real-time human-pose detection algorithm, OpenPose (Wei et al., 2016; Cao et al., 2017), implemented on each of the robot agents. Three distinct pose commands are communicated to the robot, which then moves in three different directions: left, forward, and right. These predefined commands can also be communicated to the robots as audio cues by using the Google Audio speech-to-text interface (Google LLC, 2022) running on the head substation. The real human with XR may utter any of these three predefined commands into the internal microphone of the VR headset, which is then transcribed to text. Based on the pose commands received as either visual or audio cues, the planner generates desired waypoints in the commanded direction. These waypoints are then streamed to the base control station of the real robot with XR (i.e., the real robot coupled with the robot avatar) over LAN and to the virtual robot in the environment as described in Section 3.4. The desired orientation at each set of waypoints manipulates the robots to face the human avatar in order to perceive the next gesture command. The waypoint-following policy, described in Section 3.5, is then used to calculate the control commands on each of the robot agents to reach their desired waypoint. This experiment is summarized using the schematic in Figure 9 for a virtual robot i, $i \in P$, and a real robot with XR j, $j \in \{R \cap P\}$. In this experiment, the virtual robot and the robot avatar are placed alongside each other at a fixed distance and orientation needed to perceive the human avatar in UE™. With each pose command, the robot agents move a distance of 0.5 m in the commanded direction. A total of ten pose commands are issued to each agent, and the resulting position changes are plotted in Figure 10. The results show that both the virtual robot and the robot avatar are able to correctly identify and react to all ten predefined gesture commands. Both the virtual robot and the robot avatar successfully traverse the distance with proper heading directions as shown in Figure 10. Comparison between the trajectories of the robot avatar and the virtual robot indicates that the robot avatar is able to successfully incorporate the dynamics of its real-world counterpart and hence exhibits errors induced by real-world physics, such as the effects of friction and slip, unlike the virtual robot. This experiment demonstrates how the RealTHASC facility is able to successfully simulate proximate visual interactions and incorporate real-world dynamics while providing a safe medium for human-robot collaboration.

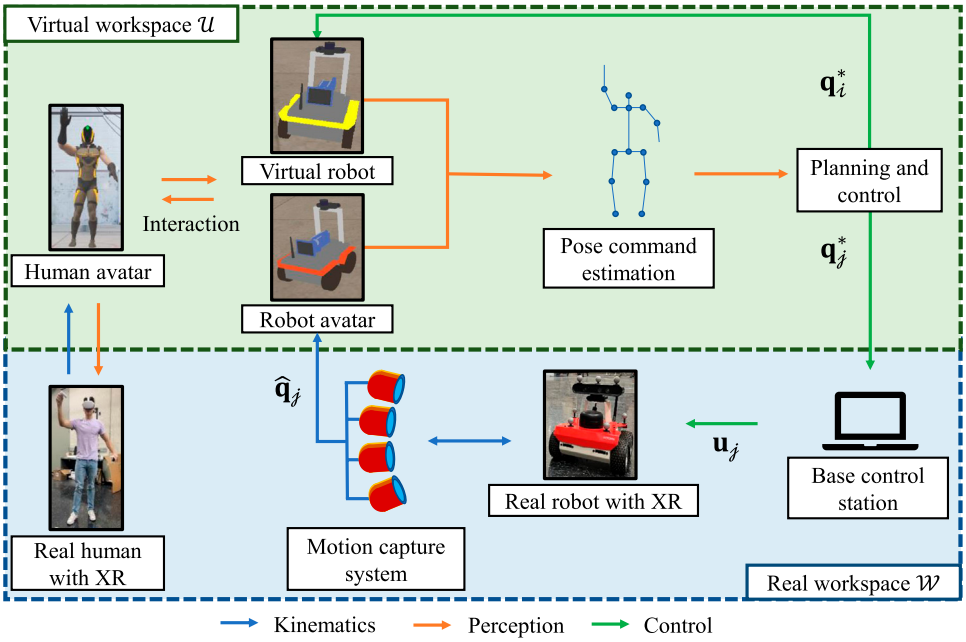
4.2 Multi-robot interaction for formation control

The second experiment is designed to illustrate closed-loop interaction and control between multiple robot agents listed in Figure 1: a virtual robot, a real robot with XR sensing in $\mathcal{U}$, and a real robot sensing in $\mathcal{W}$. The purpose of this experiment is to demonstrate that the facility is able to bridge the gap of data transfer between agents existing in simulation and the real world. In this experiment, the robot team is tasked with a leader-follower-based formation control objective. A virtual robot and robot avatar are placed into the industrial city environment created in UE™ while a real robot and a real robot with XR (i.e., real robot coupled with the robot avatar) operate in the physical lab workspace. The virtual robot is designated as the leader robot which independently moves along a pre-specified path. The real robot with XR determines its waypoints using the localization information of the virtual robot, as communicated to its avatar, while the real robot does so, in turn, by using the localization of the real robot with XR. A formation control policy is implemented onboard each robot agent to calculate the



Images were produced using Unreal Engine™ 4

FIGURE 8
First-person perspective from a virtual autonomous UGV inside the industrial city, equipped with a virtual RGB camera and implementing OpenPose for keypoint detection (top row). The robot avatar is able to recognize and interpret the manual commands provided by a real human in $\mathcal{W}$ (bottom row), namely, **(A)** left, **(B)** right, and **(C)** forward, by virtue of the human avatar created in real time using VR body tracking.



Images were produced using Unreal Engine™ 4

FIGURE 9
Human-robot collaboration is achieved by a human avatar, teleoperated by a real human with XR, commanding heading directions to the virtual robot i and robot avatar j operating in $\mathcal{U}$, using pose commands generated by the keypoint detection as shown in Figure 8.

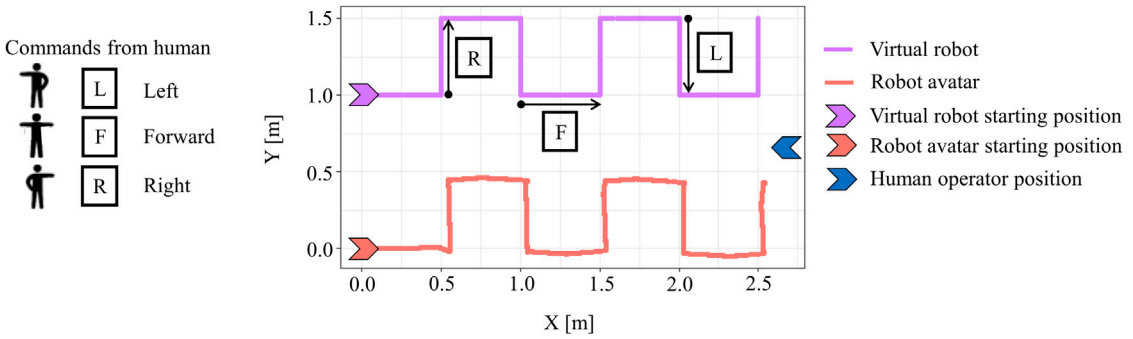


FIGURE 10
Trajectory results of the pose estimation experiment for human-robot interaction. A total of 10 pose commands are given to both the virtual robot and the robot avatar.

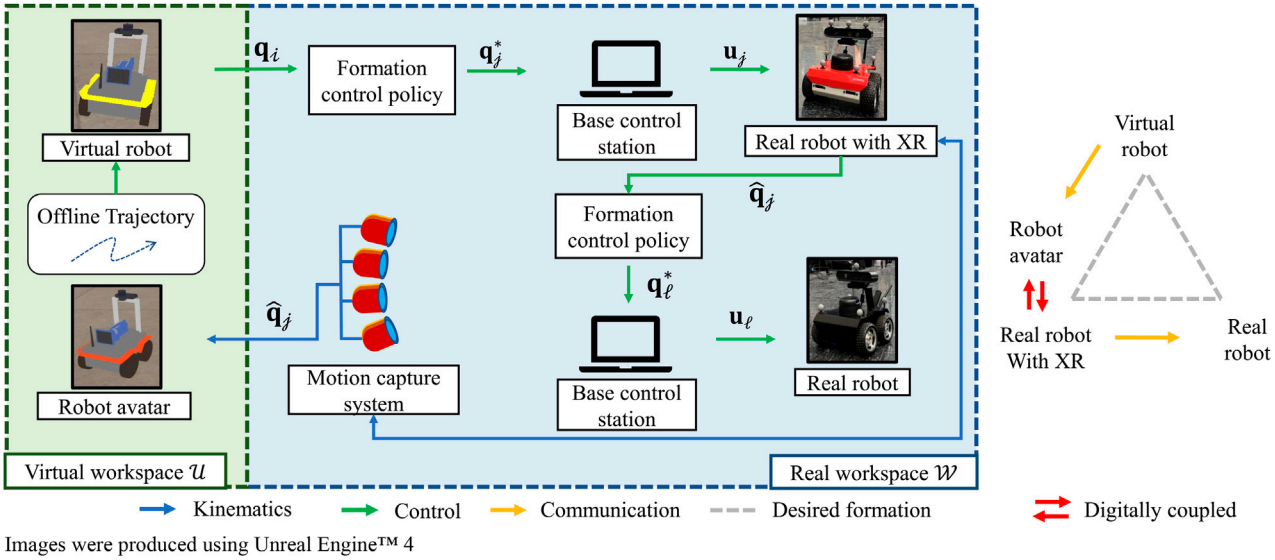


FIGURE 11
Leveraging the communication pipeline of RealTHASC, a virtual robot i, real robot with XR j with its avatar, and a real robot ℓ co-ordinate amongst themselves to maintain the isosceles triangle formation.

desired waypoints and ensure that the robot team maintains a desired formation.

The formation control experiment is illustrated in Figure 11, which features a multi-robot team comprising of a virtual robot i, $i \in P$, a real robot with XR j, $j \in \{R \cap P\}$, and a real robot ℓ, $\ell \in R$. The virtual robot in the role of a leader moves along an offline, elliptical trajectory. The state of the leader is streamed by means of a socket connection to the base control stations of the real robot with XR in $\mathcal{W}$. This base control station calculates the desired waypoint based on the state of the leader in real time to maintain an isosceles triangle formation. Simultaneously, the state of this real robot with XR is also streamed to the base control station of the real robot using socket programming through the inter-robot LAN connection, which calculates the desired waypoint for this robot to maintain the formation. This allows for decentralized formation control of a multi-robot team in and across $\mathcal{W}$ and $\mathcal{U}$. All robot agents use the policy defined in Section 3.5 to reach the desired waypoints obtained online. It is important to note that in this experiment, only the state of real robot with XR is streamed back to the virtual environment since it is the only robot with a virtual avatar.

The trajectories of the robot agents are plotted in Figure 12A. The virtual robot in UE™ follows the elliptical trajectory as designed, and the successful coupling between the real robot with XR and its virtual avatar can be observed. The robot team maintains the desired isosceles triangle formation throughout the experiment as illustrated in various instances in Figure 12A. Since the path of the leader and the desired formation have been predefined, the desired trajectories for all the robots are determined offline and the error between their positions during the experiment and these trajectories are recorded. High positional accuracy is achieved by both the agents as the largest positional error is within 0.10 m as shown in Figure 12B. This performance plot also shows that the second follower (real robot) consistently experiences lower positional accuracy when

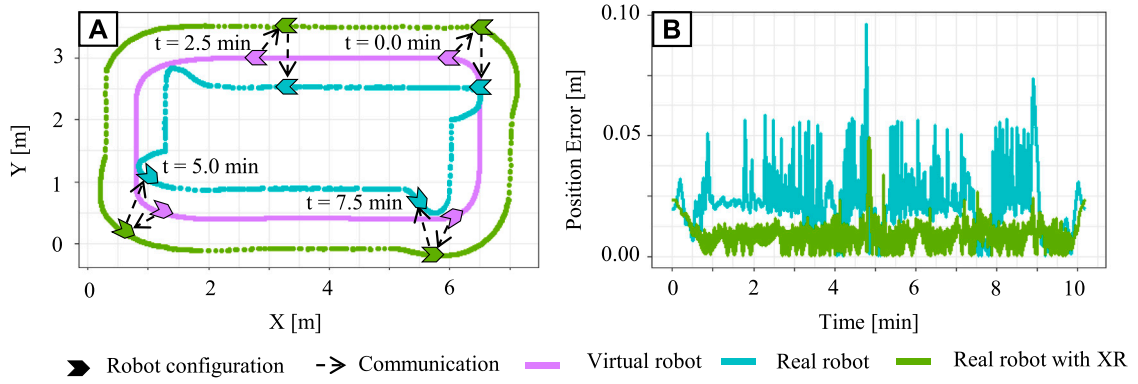


FIGURE 12
(A) Trajectory results of the leader-follower formation control experiment. The virtual robot is the leader, the real robot with XR is the first follower, and the real robot is the second follower. **(B)** The position error of each robot follower over the duration of the experiment.

compared to the first follower (real robot with XR). This is attributed to the aggregation of errors due to the decentralized nature of coordination amongst the agents. These results successfully demonstrate the capability of this testbed to establish communication between various agents existing in the real and virtual workspace and, as a result, enable real-time interaction between real and simulated agents.

4.3 Human-robot collaboration in underwater multi-target classification

This section aims to showcase a probable extension and direct application of the RealTHASC framework to intelligent vehicle systems commonly used in industrial and defense sectors, such as AUVs. Human operators collaborating with AUVs can help integrate their expertise, domain knowledge, and situational awareness, thus making these systems more robust, adaptive, and efficient. While testing underwater human-robot collaboration in the development phase can prove to be resource-intensive, testing underwater perception capability completely in simulation leads to inaccuracies when the simulation fails to incorporate environmental factors. Thus, the aim of this experiment is to augment the capabilities of the RealTHASC facility for testing collaborative multi-target classification in human-robot teams to overcome these challenges.

The undersea virtual environment hosted by RealTHASC consists of various seabed conditions as shown in Shin et al. (2022a). In this experiment, three different seabed conditions—namely sand ripples, mud, and rocks—are implemented. These seabed conditions are acquired from the UE™ Marketplace and modified using the Sculpt mode in UE™ editor to replicate the environmental conditions of the operation site. The synthetic sonar images are generated by setting a camera actor defined in UE™ to face downwards and adding a directional light from the side to mimic acoustic highlights and shadow patterns based on the orientation of the vehicle equipped with imaging sonar. Once the camera actor renders RGB images, the rendered images are then post-processed to convert RGB values into intensity values and to add realistic acoustic noises. This post-processing is conducted using MATLAB, which converts the input RGB image into a grayscale image and adds either Gaussian, speckle, or Poisson noise. An example of both an output RGB image from UE™ and post-processed synthetic sonar images are presented in Figure 13.

The synthetic images generated from this photorealistic simulation are then used to train an automatic target recognition (ATR) algorithm to classify targets in the images. In this experiment, three types of objects are used: cylinders, cubes, and spheres. A total of 1850 synthetic images are generated for the geometric targets: 650 for the cylinder, 600 for the cube, and 600 for the sphere. These images are generated with each type of object from various aspect angles to train acoustic highlight-shadow patterns, and a speckle noise with a variance of 0.1 is used in the post-processing. The ATR algorithm presented in Zhu et al. (2017), which uses a pre-trained AlexNet to extract feature vectors followed by a support vector machine (SVM) that is trained to perform the classification, is implemented in this experiment. A total of 80 images are used for training to avoid overfitting, and the remaining synthetic images are used for testing. Specifically, 21 images of a cylinder object, 27 images of a cube object, and 32 images of a sphere object are used for training the ATR algorithm. As a side note, the transfer learning performance of this ATR approach has been tested by images generated from a high-fidelity physics-based sonar simulation (Sammelmann et al., 1997) and presented in Shin et al. (2022a).

Underwater multi-target classification algorithms require multiple sonar images taken from different views to achieve a satisfactory confidence level before each object's classification is declared (Chang et al., 2018). This classification confidence level is updated based on a sensor model represented in a Bayesian network whenever a new sonar image of each object is obtained. Leveraging RealTHASC alongside the aforementioned ATR approach, a human-robot collaboration experiment is proposed as a probable extension that may be used to test the

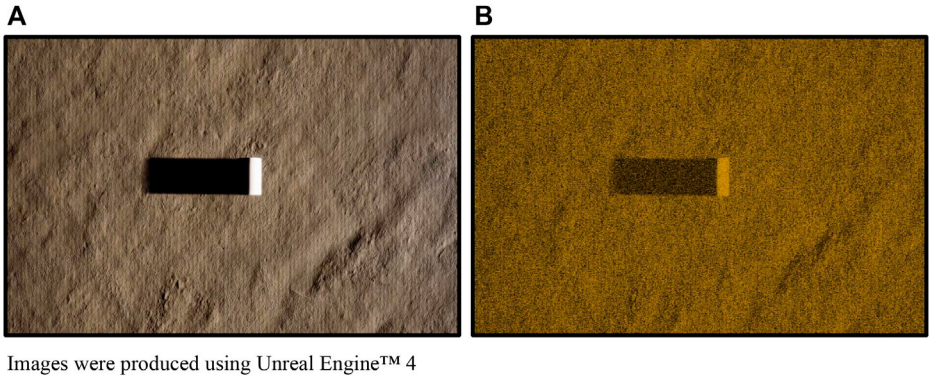


FIGURE 13
Synthetic sonar images taken from the simulation model presented in Shin et al., 2022a taken **(A)** before and **(B)** after post-processing with Gaussian white noise with mean 0 and variance of 0.05. © 2022 IEEE. Reprinted, with permission, from Shin et al. (2022a).

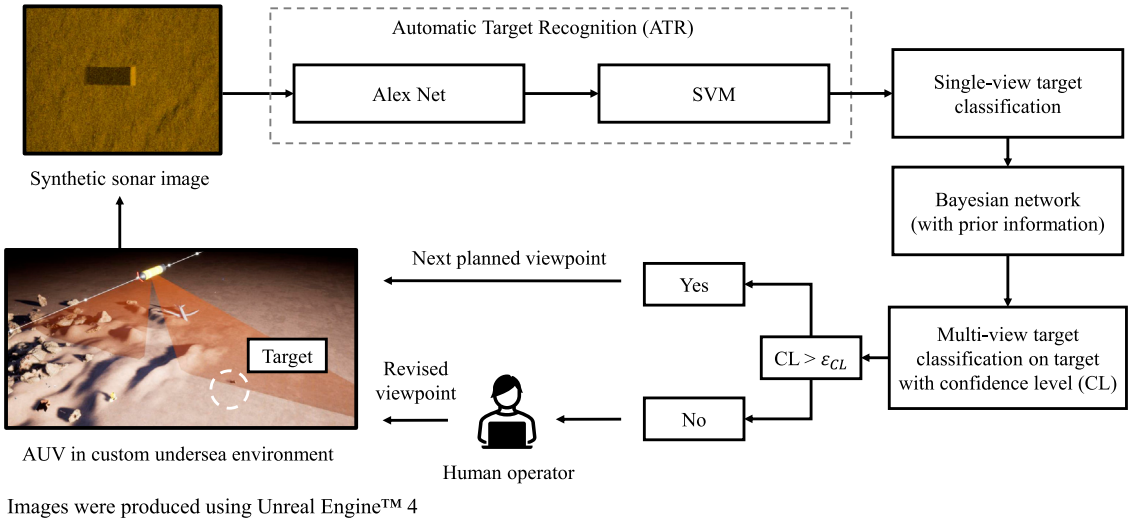


FIGURE 14
Example demonstrating human collaborating with an AUV equipped with automatic target recognition (ATR) algorithms and Bayesian network used for multi-target classification from images generated by the synthetic sonar simulator. © 2022 IEEE. Reprinted, with permission, from Shin et al. (2022a).

effect of this collaboration on multi-target classification from sonar images. The confidence level (CL) and probabilistic sensor model used in the description of this experiment are explained in further detail in Chang et al. (2018); Shin et al. (2022b). The experiment is set up such that the AUV equipped with a side-scan sonar sensor in the virtual environment uses this target classification approach while constantly communicating with a remote human operator. The human operator is notified when the CL is higher than a user-chosen threshold (ε_{CL}), which is designed to be lower than the threshold to declare target classification. This setting allows the human operator to revise the sonar images and corresponding output from the ATR algorithm before the system declares a wrong classification. Moreover, the human operator does not need to go over every false alarm. The framework of the proposed experiment is illustrated in Figure 14. This demonstrates how the RealTHASC facility can also be used for human-robot interaction applications in remote environments.

5 Conclusion

This paper presents RealTHASC, a multimodal cyber-physical XR facility that leverages state-of-the-art robotics, visualization tools, motion capture, and virtual reality technology to enable a novel experimental testbed interfacing physical and virtual worlds. Unreal Engine™ is used to create photorealistic simulated environments which facilitate interactions amongst human-autonomy teams (HATs), comprising of real agents, virtual agents, and agent avatars, tasked with achieving various objectives. These agent avatars operating in the virtual environment are teleoperated by the

real agents with XR operating in a physical environment, thus sharing real-world dynamics, while the avatars grant simulated perception for planning and decision-making. Communication pipelines enable seamless interfacing of the real and virtual workspace in order to enable real-time collaboration amongst various agents in the HATs. The results of the three experiments demonstrate the capability of this system's framework to effectively host highly flexible environments with interactive agents spanning a combination of both the real and virtual worlds. The first experiment focuses on establishing human-robot perception and effectively demonstrates closed-loop control of both virtual robots and real robots with coupled virtual avatars. Using body gestures or voice commands, the human operators effectively communicate commands with robot agents and control the trajectory of each agent in real time. With perception and control successfully established between the robots and humans in this testbed, the second experiment demonstrates that the developed facility is able to establish decentralized communication between varying robot agents. By implementing a leader-follower and formation control scenario on robot teams, this experiment effectively conveys the modality of RealTHASC to host real-time communication between the simulated and physical worlds and extends its reach to be used for multi-robot experiments. Finally, the third experiment shows the highly programmable nature of the sensors and virtual environments supported by the facility and their compatibility with the proposed human-AUV collaboration framework. Apart from such pre-deployment testing of collaboration algorithms, the RealTHASC facility can also be used as a closed-loop interaction interface to facilitate downstream tasks, such as online learning and data collection, for safety-critical applications including social navigation. Future work will extend the capabilities of this facility to include new interfaces for human operators such as haptic feedback devices and will leverage this facility to study 1) AI-supported teamwork in collaborative virtual environments, 2) decentralized AI-supported multi-agent planning and perception, 3) integration of emerging neuromorphic and insect-scale technologies, and 4) distributed sensing and control for very-large networks of agents.

Data availability statement

The raw data supporting the conclusion of this article will be made available by the authors, without undue reservation.

Ethics statement

Ethical review and approval was not required for the study on human participants in accordance with the local legislation and institutional requirements. The patients/participants provided their written informed consent to participate in this study.

Author contributions

AP, SS, and SF contributed to the overall conception and design of the study. AP and SS took the lead in writing the manuscript. JM and AP designed and implemented the human-robot perception and control experiment. MG, JM, and SS designed the formation control for the multi-robot interaction experiment. VB wrote the code for executing the waypoint-following policy for the virtual robots. KJ, CL, and JD contributed to the integration of robot avatars in the simulation environment and implemented the waypoint following policy for the real robots. SQ contributed to the design of the industrial city virtual environment. JS designed the synthetic sonar simulator and implemented the automatic target recognition algorithm. SF was the principal investigator for this research. All authors contributed to the article and approved the submitted version.

Funding

This research was funded by the Office of Naval Research Defense University Research Instrumentation Program (DURIP) grant N00014-20-S-F004. AP was supported by the Cornell Engineering Colman Fellowship, by the Alfred P. Sloan Foundation grant G-2019-11435, and by the National Science Foundation (NSF) grant EFMA-2223811. SS was supported by the Office of Naval Research (ONR) grant N00014-22-1-2513. SF was supported by the National Science Foundation (NSF) grant EFMA-2223811.

Acknowledgments

The authors would like to thank Hewenxuan Li and Qingze Huo for their valuable feedback and comments provided during the writing and review process.

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

References

Bassyouni, Z., and Elhajj, I. H. (2021). Augmented reality meets artificial intelligence in robotics: A systematic review. *Front. Robotics AI* 8, 724798. doi:10.3389/frobt.2021.724798

Bradski, G. (2000). The opencv library. *Dr. Dobb's J. Softw. Tools*.

Cao, Z., Simon, T., Wei, S.-E., and Sheikh, Y. (2017). "Realtime multi-person 2d pose estimation using part affinity fields," in Proc. of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 1302–1310. doi:10.1109/CVPR.2017.143

Chang, S., Isaacs, J., Fu, B., Shin, J., Zhu, P., and Ferrari, S. (2018). "Confidence level estimation in multi-target classification problems," in Proc. of the Detection and Sensing of Mines, Explosive Objects, and Obscured Targets XXIII (SPIE) 10628, 458–464. doi:10.1117/12.2319988

Chen, J., Gauci, M., Li, W., Kolling, A., and Groß, R. (2015). Occlusion-based cooperative transport with a swarm of miniature mobile robots. *IEEE Trans. Robotics* 31, 307–321. doi:10.1109/TRO.2015.2400731

Choi, H., Crump, C., Duriez, C., Elmquist, A., Hager, G., Han, D., et al. (2021). On the use of simulation in robotics: opportunities, challenges, and suggestions for moving forward. *Proc. Natl. Acad. Sci.* 118, e1907856118. doi:10.1073/pnas.1907856118

DeepMotion (2023). *DeepMotion SDK - virtual reality tracking*. Available at: https://www.deepmotion.com/virtual-reality-tracking.

Deitke, M., Han, W., Herrasti, A., Kembhavi, A., Kolve, E., Mottaghi, R., et al. (2020). "Robothor: an open simulation-to-real embodied ai platform," in 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 3161–3171. doi:10.1109/CVPR42600.2020.00323

Dosovitskiy, A., Ros, G., Codevilla, F., Lopez, A., and Koltun, V. (2017). "Carla: an open urban driving simulator," in Proc. of the 1st Annual Conference on Robot Learning, 1–16.

Epic Games (2019). *Unreal engine*. Available at: https://www.unrealengine.com/en-US/.

Erez, T., Tassa, Y., and Todorov, E. (2015). "Simulation tools for model-based robotics: comparison of bullet, havok, MuJoCo, ODE and PhysX," in Proc. of the 2015 IEEE International Conference on Robotics and Automation (ICRA) (IEEE), 4397–4404. doi:10.1109/ICRA.2015.7139807

Ferrari, S., and Wettergren, T. A. (2021). *Information-driven planning and control*. MIT Press.

Fong, T., Nourbakhsh, I., and Dautenhahn, K. (2003). A survey of socially interactive robots. *Robotics Aut. Syst.* 42, 143–166. doi:10.1016/S0921-8890(02)00372-X

Garg, G., Kuts, V., and Anbarjafari, G. (2021). Digital twin for fanuc robots: industrial robot programming and simulation using virtual reality. *Sustainability* 13, 10336. doi:10.3390/su131810336

Gemerek, J., Ferrari, S., Wang, B. H., and Campbell, M. E. (2019). Video-guided camera control for target tracking and following. *IFAC-PapersOnLine* 51, 176–183. doi:10.1016/j.ifacol.2019.01.062

Google LLC (2022). *Google cloud speech API*. Available at: https://cloud.google.com/speech-to-text/docs/.

Guerra, W., Tal, E., Murali, V., Ryou, G., and Karaman, S. (2019). "Flightgoggles: photorealistic sensor simulation for perception-driven robotics using photogrammetry and virtual reality," in Proc. of the 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE), 6941–6948. doi:10.1109/IROS40897.2019.8968116

Hu, J., and Lanzon, A. (2018). An innovative tri-rotor drone and associated distributed aerial drone swarm control. *Robotics Aut. Syst.* 103, 162–174. doi:10.1016/j.robot.2018.02.019

Inamura, T., and Mizuchi, Y. (2021). Sigverse: A cloud-based vr platform for research on multimodal human-robot interaction. *Front. Robotics AI* 8, 549360. doi:10.3389/frobt.2021.549360

Kao, W.-W. (1991). "Integration of gps and dead-reckoning navigation systems," in *Proc. Of the vehicle navigation and information systems conference, 1991* (IEEE), 2, 635–643. doi:10.1109/VNIS.1991.205808

Koenig, N., and Howard, A. (2004). "Design and use paradigms for gazebo, an open-source multi-robot simulator," in *Proc. Of the 2004 IEEE/RSJ international conference on intelligent robots and systems (IROS)(IEEE cat. No. 04CH37566)* (IEEE), 3, 2149–2154. doi:10.1109/IROS.2004.1389727

Krajník, T., Nitsche, M., Faigl, J., Vaněk, P., Saska, M., Přeučil, L., et al. (2014). A practical multirobot localization system. *J. Intelligent Robotic Syst.* 76, 539–562. doi:10.1007/s10846-014-0041-x

Liu, R., Natarajan, M., and Gombolay, M. C. (2021). Coordinating human-robot teams with dynamic and stochastic task proficiencies. *ACM Trans. Human-Robot Interact. (THRI)* 11, 1–42. doi:10.1145/3477391

Michel, O. (2004). Cyberbotics ltd. Webots™: professional mobile robot simulation. *Int. J. Adv. Robotic Syst.* 1, 5. doi:10.5772/5618

Mizuchi, Y., and Inamura, T. (2017). "Cloud-based multimodal human-robot interaction simulator utilizing ROS and Unity frameworks," in *2017 IEEE/SICE international symposium on system integration (SII)* (IEEE), 948–955. doi:10.1109/SII.2017.8279345

Murnane, M., Higgins, P., Saraf, M., Ferraro, F., Matuszek, C., and Engel, D. (2021). "A simulator for human-robot interaction in virtual reality," in 2021 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW) (IEEE), 470–471. doi:10.1109/VRW52623.2021.00117

Naghsh, A. M., Gancet, J., Tanoto, A., and Roast, C. (2008). "Analysis and design of human-robot swarm interaction in firefighting," in *Proc. Of RO-MAN 2008-the 17th IEEE international symposium on robot and human interactive communication* (IEEE), 255–260. doi:10.1109/ROMAN.2008.4600675

Nourbakhsh, I. R., Sycara, K., Koes, M., Yong, M., Lewis, M., and Burion, S. (2005). Human-robot teaming for search and rescue. *IEEE Pervasive Comput.* 4, 72–78. doi:10.1109/MPRV.2005.13

Ognibene, D., Foulsham, T., Marchegiani, L., and Farinella, G. M. (2022). Editorial: active vision and perception in human-robot collaboration. *Front. Neurorobotics* 16, 848065. doi:10.3389/fnbot.2022.848065

Oh, J., Howard, T. M., Walter, M. R., Barber, D., Zhu, M., Park, S., et al. (2017). "Integrated intelligence for human-robot teams," in *Proc. Of the 2016 international symposium on experimental robotics* (Springer), 309–322. doi:10.1007/978-3-319-50115-4_28

Pendleton, S. D., Andersen, H., Du, X., Shen, X., Meghjani, M., Eng, Y. H., et al. (2017). Perception, planning, control, and coordination for autonomous vehicles. *Machines* 5, 6. doi:10.3390/machines5010006

Puig, X., Ra, K., Boben, M., Li, J., Wang, T., Fidler, S., et al. (2018). "Virtualhome: simulating household activities via programs," in 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 8494–8502. doi:10.1109/CVPR.2018.00886

Qiu, W., and Yuille, A. (2016). "Unrealcv: connecting computer vision to unreal engine," in *Proc. Of the computer vision – ECCV 2016 workshops* (Springer International Publishing), 909–916. doi:10.1007/978-3-319-49409-8_75

Reiners, D., Davahli, M. R., Karwowski, W., and Cruz-Neira, C. (2021). The combination of artificial intelligence and extended reality: A systematic review. *Front. Virtual Real.* 2, 721933. doi:10.3389/frvir.2021.721933

Sammelmann, G. S., Fernandez, J. E., Christoff, J. T., Vaizer, L., Lathrop, J. D., Sheriff, R. W., et al. (1997). "High-frequency/low-frequency synthetic aperture sonar," in Proc. of the Detection and Remediation Technologies for Mines and Minelike Targets II (SPIE) 3079, 160–171. doi:10.1117/12.280850

Shah, S., Dey, D., Lovett, C., and Kapoor, A. (2018). "Airsim: high-fidelity visual and physical simulation for autonomous vehicles," in Proc. of the Field and Service Robotics: Results of the 11th International Conference (Springer), 621–635. doi:10.1007/978-3-319-67361-5_40

Shen, B., Xia, F., Li, C., Martín-Martín, R., Fan, L., Wang, G., et al. (2021). "Igibson 1.0: A simulation environment for interactive tasks in large realistic scenes," in 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 7520–7527. doi:10.1109/IROS51168.2021.9636667

Shin, J., Chang, S., Bays, M. J., Weaver, J., Wettergren, T. A., and Ferrari, S. (2022a). "Synthetic sonar image simulation with various seabed conditions for automatic target recognition," in *Proc. Of the OCEANS 2022, hampton roads* (IEEE), 1–8. doi:10.1109/OCEANS47191.2022.9977275

Shin, J., Chang, S., Weaver, J., Isaacs, J. C., Fu, B., and Ferrari, S. (2022b). Informative multiview planning for underwater sensors. *IEEE J. Ocean. Eng.* 47, 780–798. doi:10.1109/JOE.2021.3119150

Škulj, G., Malus, A., Kozjek, D., Selak, L., Bračun, D., Podržaj, P., et al. (2021). An architecture for sim-to-real and real-to-sim experimentation in robotic systems. *Procedia CIRP* 104, 336–341. doi:10.1016/j.procir.2021.11.057

SpurnŸ, V., Báča, T., Saska, M., Pěnička, R., Krajník, T., Thomas, J., et al. (2019). Cooperative autonomous search, grasping, and delivering in a treasure hunt scenario by a team of unmanned aerial vehicles. *J. Field Robotics* 36, 125–148. doi:10.1002/rob.21816

Wei, S.-E., Ramakrishna, V., Kanade, T., and Sheikh, Y. (2016). "Convolutional pose machines," in Proc. of the IEEE conference on Computer Vision and Pattern Recognition, 4724–4732. doi:10.1109/CVPR.2016.511

Zhu, P., Isaacs, J., Fu, B., and Ferrari, S. (2017). Deep learning feature extraction for target recognition and classification in underwater sonar images," in Proc. of the 2017 IEEE 56th annual conference on decision and control (CDC). IEEE, 2724–2731. doi:10.1109/CDC.2017.8264055