applied sciences

Article

A Multi-Robot Spatial Augmented Reality Sandbox with Virtual Sensors for Computational Thinking Education

Jesús Lárez Mata [1], Pablo Garaizar [2,*] and Andoni Eguíluz [2]

[1] School of Computer Engineering, Universidad Católica Andrés Bello (UCAB), Av. Teherán, Urb. Montalbán, Parroquia La Vega, Ap. 20332, Caracas 1020, Venezuela; jlarez@ucab.edu.ve
[2] Faculty of Engineering, University of Deusto, Av. Universidades, 24, 48007 Bilbao, Spain; andoni.eguiluz@deusto.es
* Correspondence: garaizar@deusto.es

Featured Application

The proposed Spatial Augmented Reality (SAR) multi-robot arena constitutes a cost-effective hybrid platform for both educational use and applied research in distributed robotics. In educational contexts, the platform can support progressive learning pathways in computational thinking, concurrent programming, and multi-agent coordination. From an applied perspective, the environment can function as a rapid prototyping and validation testbed for distributed algorithms, hybrid physical–virtual control approaches, and human–robot interaction techniques. Potential application domains include preliminary evaluation of coordination mechanisms in logistics robotics, cooperative inspection scenarios, smart manufacturing processes, and environmental monitoring systems.

Abstract

In this paper, we introduce an immersive Spatial Augmented Reality (SAR) arena where multiple mobile robots interact with virtual sensors and actuators, enabling novice students to explore distributed and parallel algorithms. The arena uses overhead projection to display virtual objects (e.g., walls, targets, and sensor fields) onto a physical workspace, while robots (Smart Cutebot controlled by Micro:bit) operate under both physical and projected inputs. This framework is implemented via specialized software extensions for Microsoft MakeCode, offering varying levels of abstraction suitable for both novice learners and advanced researchers. The system's effectiveness is validated through technical characterization of its positioning sensors (achieving an accuracy of over 99% and a precision margin of ± 0.1 cm) and qualitative user studies with primary and university-level students. The findings suggest that the virtualization of robotic components through SAR significantly lowers the entry barrier for understanding distributed programming and collective robotics while fostering the development of computational thinking through immersive, collaborative interaction.

Keywords: educational robotics; spatial augmented reality (SAR); virtual sensors/actuators; distributed programming; computational thinking

Academic Editors: Kristian Dokic and Mladen Bošnjaković

Received: 25 March 2026
Revised: 17 April 2026
Accepted: 20 April 2026
Published: 23 April 2026

1. Introduction

Robotics has become one of the most rapidly evolving technological domains, gaining a growing influence on industrial productivity, economic development, and everyday human activities. In parallel with these advances, robotics has progressively entered educational contexts, where it is used both as a subject of study and as a pedagogical tool. Educational robotics has shown strong potential to promote engagement in Science, Technology, Engineering, and Mathematics (STEM) by enabling learners to design, construct, and program robotic artifacts in meaningful and motivating learning experiences.

Within educational environments, robotics can fulfill multiple roles: it can serve as an object of learning, as a didactic medium for introducing computational concepts, and as technological support for teaching practice. In particular, micro-robots have been widely adopted to introduce programming and computational thinking at primary and secondary education levels, while simultaneously functioning as experimental platforms for advanced robotics research in higher education.

However, many relevant robotic applications cannot be effectively addressed by a single autonomous system. Limitations in sensing, actuation, spatial coverage, robustness, and cost efficiency have motivated the development of multi-robot systems (MRS), in which multiple agents cooperate to achieve shared objectives. Depending on the coordination strategies employed, these systems can be framed under different paradigms, including centralized or decentralized architectures, multi-agent robotics (MAR) approaches based on local decision-making and communication, and swarm-based robotics (SBR), which rely on emergent collective behavior and stigmergic coordination mechanisms [1,2].

Despite their advantages, multi-robot solutions introduce challenges typically associated with distributed systems, such as communication reliability, synchronization, data consistency, and fault tolerance [3]. These complexities constitute a significant barrier for novice learners, particularly in educational settings where the primary goal is the development of computational thinking skills rather than mastery of low-level implementation details [4].

To address these difficulties, several programming environments have been proposed to introduce distributed and concurrent computing concepts in accessible ways. For instance, NetsBlox extends visual programming paradigms with abstractions such as peer-to-peer message passing and remote procedure calls, allowing learners to develop distributed applications such as multiplayer games [5]. Although such environments reduce the cognitive load associated with distributed programming, their integration with physical multi-robot educational platforms remains limited.

The practical deployment of educational robotics systems also faces persistent constraints related to cost, maintenance, hardware availability, and configurability. Robots are inherently complex cyber–physical systems that combine mechanical, electronic, and software components, often requiring multidisciplinary expertise for effective use. These challenges become more pronounced in multi-robot scenarios, especially when specialized sensors or actuators are needed. Even when financial resources are available, institutions frequently encounter difficulties in acquiring suitable hardware or adapting existing platforms to specific pedagogical or experimental requirements.

In highly specialized domains such as swarm intelligence, implementing mechanisms like stigmergic coordination through environmental modification is particularly difficult in physical settings due to sensing and interaction limitations. Consequently, simulation environments are commonly used to prototype and evaluate coordination strategies prior to real-world deployment. Nevertheless, simulations cannot fully reproduce the uncertainty, noise, and embodiment constraints that characterize physical robotic systems.

From a pedagogical standpoint, educational robotics initiatives at primary and secondary levels typically prioritize the development of computational thinking rather than robot construction as an end in itself. This perspective is rooted in constructionist learning theory [6], which emphasizes learning through the creation of meaningful artifacts. Papert's work with the LOGO programming language and the Turtle robot demonstrated how computers can function as meta-tools that allow learners to explore powerful ideas through concrete experiences. Later developments such as the Scratch programming environment introduced key design principles for learning technologies (low floors, high ceilings, and wide walls) to ensure accessibility, scalability, and creative diversity [7].

Although existing robotics platforms contribute to these pedagogical goals, achieving these three design principles simultaneously remains challenging. Hardware constraints can limit accessibility, while low-level programming requirements may reduce learner engagement. Moreover, few platforms provide a coherent progression from introductory programming activities to more advanced topics such as distributed coordination, multi-agent interaction, or collective behavior.

Recent advances in immersive technologies (including Virtual Reality, Augmented Reality, and Mixed Reality) offer new opportunities to enhance robotics learning experiences through novel forms of visualization and interaction [8,9]. In particular, Spatial Augmented Reality, which integrates virtual elements directly into physical environments through projection, enables shared and device-free collaborative interaction [10]. Previous research has explored AR-based interaction with robotic systems for monitoring, control, and analysis [11–15]. However, the combination of SAR with low-cost physical multi-robot platforms and block-based distributed programming environments remains largely unexplored.

To address this gap, this paper presents a low-cost Spatial Augmented Reality arena designed to support programming activities involving multiple physical robots across educational levels. The proposed platform integrates real and virtual entities (including robots, obstacles, and manipulable resources) through computer vision tracking and projected interactive artifacts. A multi-layer software architecture provides programming interfaces at different levels of abstraction, allowing learners to focus on coordination strategies and problem solving rather than low-level technical details.

The system is implemented using Smart Cutebot robots controlled by BBC micro:bit V2 boards and programmed through custom extensions developed for the Microsoft MakeCode environment, which supports both block-based visual programming and textual coding in Python (v. 3.x) and JavaScript (ES 2015).

The main goal of this research is to design and validate a low-cost Spatial Augmented Reality multi-robot platform that integrates virtual sensing, immersive visualization, and progressive programming abstractions to support the learning of distributed robotics concepts. The study evaluates both the technical performance of the system and its educational usability, while also considering the potential influence of novelty effects on learner engagement.

We hypothesize that integrating Spatial Augmented Reality with virtual sensing and multi-level programming abstractions can reduce the complexity of distributed robotics and enable novice learners to engage with and understand multi-agent coordination concepts, although observed engagement may be partially influenced by the novelty of the immersive environment.

The main contributions of this work are: (a) The design and implementation of a low-cost Spatial Augmented Reality multi-robot arena suitable for educational contexts; (b) A multi-layer programming framework that enables progressive abstraction from individual robot control to distributed multi-agent coordination; (c) The introduction of virtual sensing and actuation mechanisms that extend robot capabilities beyond physical hardware limitations; (d) The integration of distributed robotics concepts into a block-based programming ecosystem, facilitating early exposure to concurrent and distributed computational models; and (e) A preliminary usability validation involving novice learners performing collaborative multi-robot tasks.

2. Related Work

2.1. Educational Robotics and Computational Thinking

Educational robotics has become a widely adopted approach for fostering computational thinking, problem-solving abilities, and engagement in STEM learning. Constructionist learning theory established the conceptual foundations for this approach by emphasizing learning through the design and programming of meaningful artifacts rather than through passive instruction [6]. These principles were later operationalized in block-based programming environments such as Scratch, which introduced the design guidelines of low floors, high ceilings, and wide walls to support accessibility, scalability, and creative exploration [7].

Several systematic reviews have highlighted the positive educational impact of robotics activities. Early meta-analyses reported improvements in motivation and conceptual understanding across multiple educational levels [16,17], while later systematic reviews confirmed their effectiveness in promoting computational thinking and collaborative learning [18]. More recent work has emphasized the importance of tangible interaction in early programming education, highlighting how physical robots can provide concrete representations of abstract computational processes and support embodied cognition [9,19].

Despite these benefits, most educational robotics platforms focus on individual robot programming tasks involving motion control, sensor integration, or reactive behaviors. While such activities are effective for introductory learning, they rarely expose students to contemporary robotics challenges such as distributed coordination, asynchronous communication, or collective decision-making [20,21]. Consequently, a persistent gap remains between entry-level robotics learning experiences and the distributed, networked nature of modern robotic systems.

2.2. Multi-Robot and Multi-Agent Robotics Platforms for Education

Research in multi-robot systems and multi-agent robotics has produced extensive frameworks for coordination, task allocation, and collective behavior. Foundational work in swarm intelligence demonstrated how decentralized control and stigmergic communication mechanisms can enable robust group behavior inspired by biological systems [1]. Subsequent research in distributed robotics has explored scalable coordination strategies and emergent behaviors across diverse application domains [2].

Although these advances have significantly influenced robotics research, their integration into educational platforms has been limited. Some programming environments have attempted to introduce distributed computing concepts through high-level abstractions. For example, NetsBlox incorporates message passing and remote procedure calls into block-based programming, enabling learners to create distributed applications such as multiplayer games [5]. Similarly, layered programming approaches have been proposed for mobile robotics to support progressive transitions from low-level motor control to behavioral programming [22].

However, relatively few educational systems enable learners to experiment with coordinated behavior among multiple physical robots within accessible programming environments. Existing solutions often rely on simulation-based agents, impose significant technical setup requirements, or provide limited support for embodied interaction. Platforms explicitly designed for swarm robotics education with tangible robots remain scarce, despite the growing importance of collective robotics in research and industry.

2.3. Immersive Technologies for Robotics Interaction and Learning

Immersive technologies, including Virtual Reality (VR), Augmented Reality (AR), and Mixed Reality (MR), have increasingly been explored as tools for enhancing robotics interaction, training, and education. Reviews of extended reality applications in learning environments highlight their potential to improve conceptual understanding by linking abstract representations to physical phenomena and enabling experiential learning [23,24].

In robotics research, AR interfaces have been used to support monitoring, debugging, and coordination of robot teams by overlaying virtual information onto physical environments or video feeds. Early work demonstrated how augmented visualizations can facilitate comprehension of multi-robot coordination processes [11,13,14], while mixed-reality interaction paradigms have been proposed to extend robot capabilities through virtual obstacles, goals, or environmental constraints [12,25]. More recent surveys of AR-enhanced human–robot interaction categorize design approaches involving spatial cues, virtual agents, and state visualization mechanisms that support situational awareness and collaborative control [15,26,27].

A recent systematic review specifically examining the integration of AR and robotics in education reports that AR-enhanced robotics activities can significantly increase student engagement, motivation, enthusiasm, and performance [9,28]. However, the review also highlights that large-scale classroom adoption remains limited due to cost, technical complexity, and the lack of scalable pedagogical frameworks. Furthermore, most existing studies focus on single-robot interaction scenarios or visualization support, with comparatively little attention to programming-centric learning activities involving multiple coordinated robots [14,25].

Spatial Augmented Reality, which projects virtual elements directly onto physical environments without requiring wearable displays, represents a particularly promising approach for collaborative robotics education [10,29]. By supporting shared visualization spaces and embodied interaction, SAR can facilitate group learning activities and provide intuitive representations of distributed system states. Nevertheless, prior work has predominantly explored SAR for teleoperation, task visualization, or performance analysis rather than as an integrated environment for multi-robot programming and coordination learning.

Importantly, the literature reveals a notable missing category: the use of immersive technologies (particularly AR or SAR) to support tangible multi-robot swarm learning experiences. While AR interfaces for robot teams have been studied in research contexts, their adaptation to low-cost educational ecosystems remains underexplored.

2.4. Virtual Sensing, Actuation, and Hybrid Physical–Virtual Environments

Hybrid experimental environments that combine physical robots with virtual entities have long been investigated as a means to increase flexibility and reduce hardware constraints. Concepts related to virtual sensing (in which robots obtain environmental information from external perception systems or computational models rather than onboard sensors) have been proposed in robotics middleware and distributed perception architectures [29]. These approaches enable the extension of robot capabilities without modifying hardware and are particularly relevant in swarm robotics scenarios requiring shared environmental representations for indirect coordination mechanisms such as stigmergy.

Simulation environments have traditionally played a central role in robotics research and education by enabling safe, scalable, and reproducible experimentation. However, purely simulated systems cannot fully reproduce the uncertainty, latency, and embodiment constraints that characterize real robotic platforms. As a result, hybrid physical–virtual environments have emerged as a promising compromise, combining the realism of physical robots with the configurability of virtual artifacts.

Despite these advances, low-cost educational platforms that simultaneously integrate virtual sensing and actuation mechanisms, spatially immersive interfaces, and support for programming coordinated behaviors among multiple physical robots remain largely absent. Existing solutions tend to address these dimensions independently rather than within unified pedagogical ecosystems.

2.5. Research Gap

The literature reviewed above reveals three largely independent research trajectories: (a) Educational robotics platforms that effectively support introductory programming but rarely address distributed coordination or collective robotics; (b) Multi-robot and swarm robotics research platforms that enable advanced experimentation but often require substantial technical expertise and lack accessibility for novice learners; and (c) Immersive and augmented reality interfaces that enhance visualization and interaction yet are seldom integrated into programming-centered robotics learning environments.

This fragmentation indicates a clear need for integrated platforms that allow learners to explore distributed robotics concepts through embodied interaction, progressive abstraction, and immersive visualization [14]. The platform proposed in this work addresses this gap by combining low-cost physical multi-robot interaction, Spatial Augmented Reality visualization, virtual sensing and actuation mechanisms, and a multi-layer programming framework embedded within a block-based educational ecosystem. This approach aims to support a continuous learning trajectory from introductory computational thinking activities to advanced topics in multi-agent coordination and collective robotics.

3. Architecture

This section describes the design of the proposed Spatial Augmented Reality multi-robot platform. The architecture is organized into six complementary perspectives: (1) hardware architecture, (2) software architecture, (3) communications, (4) virtual sensor/actuator design, (5) robot behavior primitives, and (6) high-level programming interface (sandbox). This decomposition clarifies how immersive visualization, hybrid physical–virtual interaction, and distributed programming support are integrated into a single educational robotics ecosystem.

3.1. Hardware Architecture

The Spatial Augmented Reality Arena constitutes the physical workspace in which multiple robots interact with real and virtual entities. The arena is delimited using fiducial markers that define a shared coordinate system for perception and projection. A zenithal camera (Microsoft LifeCam Full 1080 HD, Microsoft, Redmond, WA, USA) captures the state of the environment, while a projector (Viewsonic VS16907, Viewsonic, Walnut, CA, USA) overlays virtual elements directly onto the physical surface.

The hardware configuration includes a projection surface defining the interaction workspace, a ceiling-mounted camera for computer vision tracking, a projector for SAR visualization, a gateway computer managing communication and world state, micro-robots operating within the arena, and robot programming computers (see Figure 1).

Using physical robots is essential to expose learners to uncertainties inherent to embodied systems, such as sensor noise, actuation errors, and communication latency. These characteristics are difficult to reproduce in purely simulated environments yet play a crucial role in understanding real distributed robotic behavior.

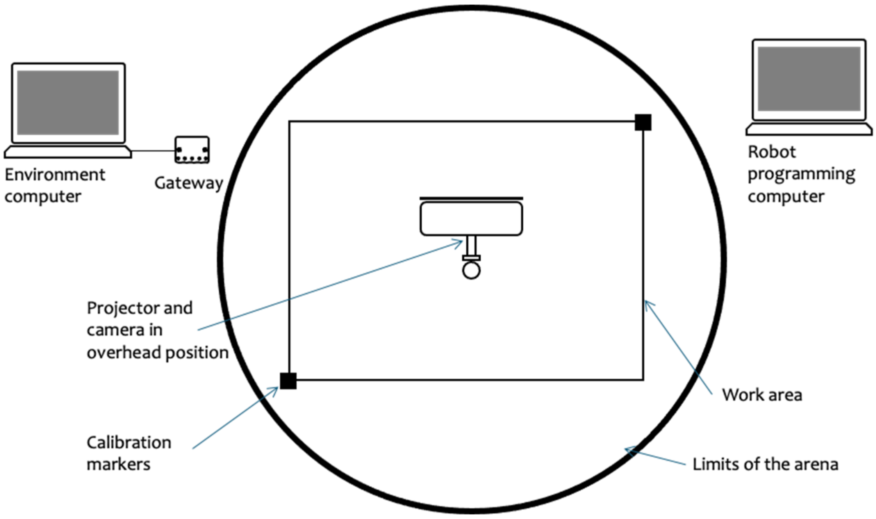


Figure 1. Main components of the Spatial Augmented Reality Arena.

The environment integrates both real entities (robots, physical obstacles) and virtual entities projected onto the arena (see Figure 2). Virtual elements may be static (including areas, virtual obstacles, paths, or symbolic markers), temporary (appearing or disappearing based on events), or mobile (e.g., manipulable resources that can be transported by one or multiple robots). Status indicators projected onto virtual objects provide immediate feedback on collaborative actions (e.g., when multiple robots are required to transport a large resource).

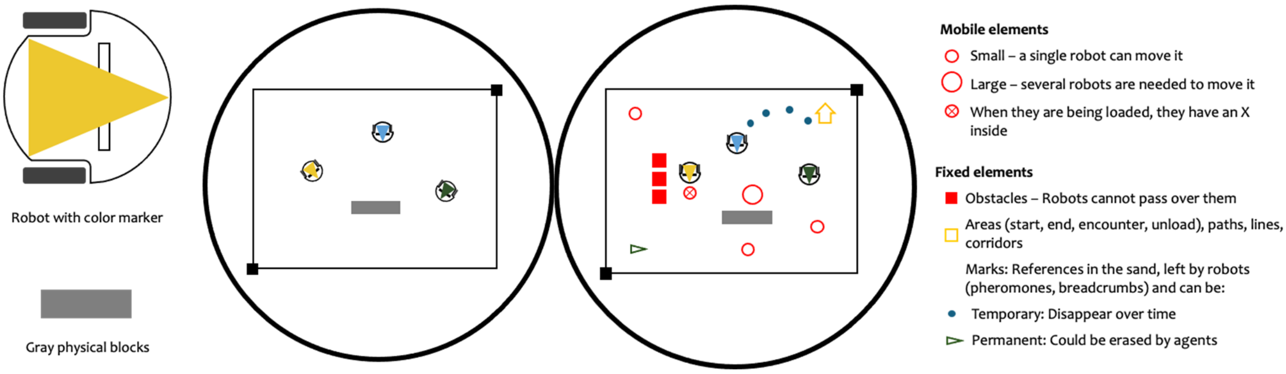


Figure 2. (**a**) Robots and physical obstacles; (**b**) Virtual elements.

3.2. Software Architecture

The platform software follows a layered distributed architecture [3] designed to support progressive abstraction in multi-robot programming. A Hardware Abstraction Layer (HAL) isolates robot-specific details, enabling portability across different robotic platforms.

On the environment side, the main software components include a communication gateway managing wireless message routing, a computer vision module estimating robot position and orientation and detecting entities, a SAR visualization engine responsible for real-time projection, a world model manager maintaining a consistent representation of physical and virtual entities, an event handler coordinating asynchronous interactions, a virtual sensor/actuator module and a high-level programming interface supporting centralized or hybrid coordination logic (see Figure 3, left).

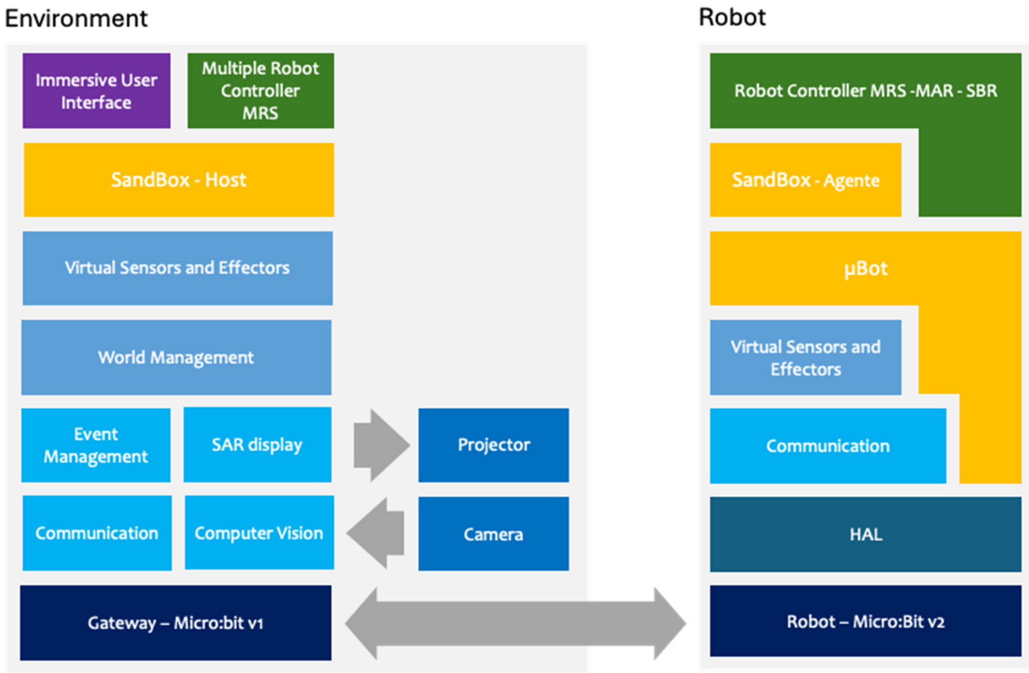


Figure 3. Software architecture of the proposed environment (**left**) and robots (**right**).

On the robot side, the software stack includes HAL drivers for sensors and actuators, a communication module for distributed message exchange, virtual sensor/actuator interfaces, and multi-level programming APIs enabling implementation of controllers for multi-robot systems, multi-agent robotics, and swarm robotics scenarios (see Figure 3, right).

Applications are developed using event-driven concurrent programming, combining asynchronous actions with blocking synchronization primitives. This model allows learners to experiment with distributed coordination patterns such as help requests, leader following, or collaborative transport while abstracting low-level networking and synchronization mechanisms.

3.3. Communications, Computer Vision and SAR Display

Communication among robots and between robots and the environment is implemented using wireless message passing mediated by a gateway. Logical communication range can be configured independently of physical connectivity, allowing the definition of proximity-based interaction constraints typical of swarm robotics experiments (see Figure 4).

The platform supports unicast communication between individual agents, multicast and broadcast communication within dynamically defined groups, and anycast interaction patterns, enabling agents to request assistance from nearby peers. Messages use compact fixed-field formats (Sender ID: 1 char, Receiver ID: 1 char, Command: 2 char, Parameters: 15 char) to reduce transmission overhead and ensure predictable latency. Range-limited logical communication enables experimentation with decentralized coordination strategies based on local perception and interaction.

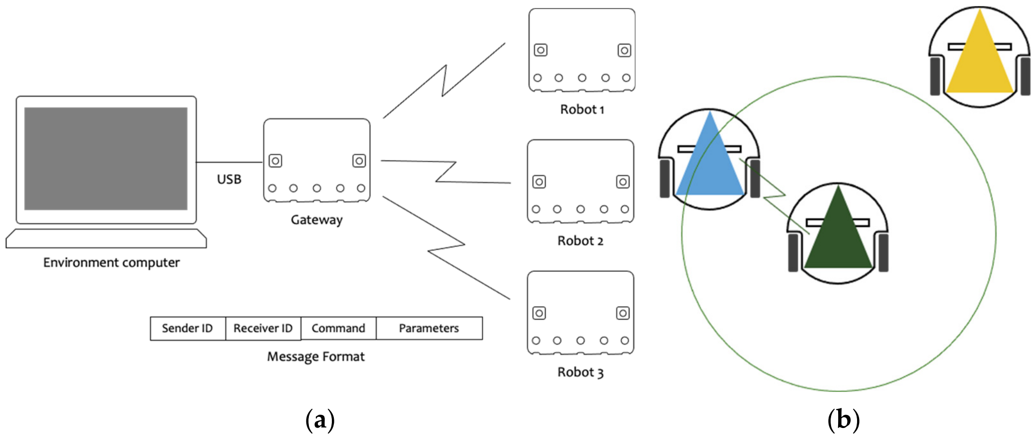


Figure 4. Communication in the distributed environment: (**a**) Wireless communication between environmental elements and message format; (**b**) Communication within a range between different robots (blue and green robots are within range, whilst yellow robot is out of range).

This layer is supported by a custom Spatial Augmented Reality (SAR) visualization engine developed in Python (v. 3.x), which leverages PySimpleGUI (v. 5.0) for real-time projection of virtual elements onto the physical arena (see Figure 5). This engine is tightly integrated with the OpenCV-based perception pipeline, enabling synchronized updates between the detected world state and the projected visual feedback. Rather than relying on external game engines, this lightweight architecture ensures low-latency interaction and simplifies deployment in educational settings while maintaining sufficient flexibility to represent dynamic multi-robot scenarios.

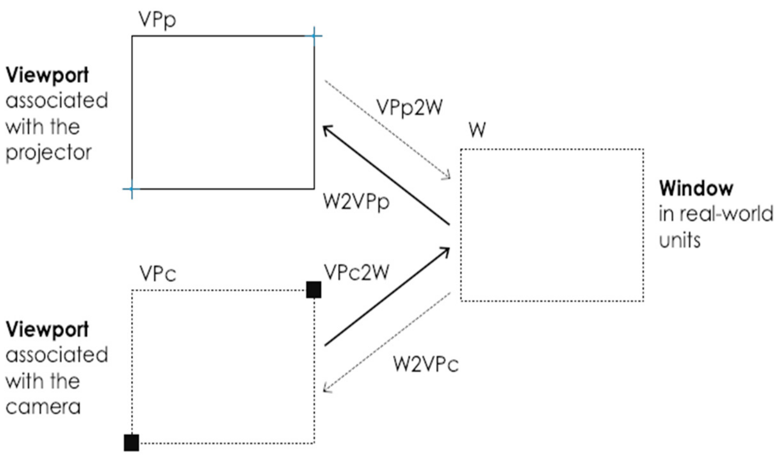


Figure 5. Primitives for the transformations between the viewports of the camera (black squares represent the physical limits of the arena), the projector and the window in real-world units.

3.4. Virtual Sensors and Actuators

We provide each robot with software-based (i.e., "virtual") sensors and actuators, spatially projected via the SAR system, to simulate capabilities such as absolute localization, collision detection, proximity sensing, or object manipulation that may not be available through onboard hardware.

Virtual sensing is implemented by combining computer vision tracking with computational models of the environment. This approach allows robots to access information such as absolute position and orientation in the arena, distances to real or virtual entities, detection of arena boundaries or restricted zones, or perception of symbolic markers or virtual signals.

These mechanisms extend earlier concepts of virtual sensing in robotics middleware and distributed perception architectures, where environmental variables are inferred through external sensing infrastructure rather than onboard devices. In immersive robotics interfaces, AR-based overlays have similarly been used to provide robots and human operators with enhanced situational awareness (e.g., AR-supported robot monitoring systems).

Virtual actuators operate on projected entities and enable actions such as picking and placing virtual resources, placing or removing markers, activating virtual buttons or displays, and collaboratively transporting large virtual objects.

Because virtual actuators act exclusively on virtual elements, they enable the exploration of complex coordination behaviors without requiring specialized mechanical hardware. Recent educational research has highlighted the pedagogical value of such abstractions, showing that virtual sensors can function as scaffolding mechanisms that facilitate learning progression toward advanced distributed robotics concepts [25].

This hybrid physical–virtual interaction model preserves the realism of embodied motion while providing flexible experimentation capabilities typical of simulation environments.

3.5. Robot Behavior Primitives

User applications are programmed using the concurrent programming paradigm, managing different threads, as well as programming based on event dispatching and asynchronous functions. Figure 6 shows how two agents (Alice and Bob) collaborate to transport a large mobile object between them to a zone in the arena called "home." The sandbox generates events that trigger threads in the agents and agents can generate events that trigger threads in the same agent and/or other agents. In this example, the agent Alice finds an object and requests help from any other agent (using anycast communication). Event-driven programming blocks (e.g., <On Help Arrives>, <On arrived home>, <On arrived to help>, <On "Follow me" received>) trigger execute primitives (e.g., <Take object between various>, <Follow me>, <Go home>, <Drop Load>, <Follow leader>). Some of them are synchronous with blocking behavior (e.g., <Take object between various>, where the action is completed before proceeding), and others are asynchronous (e.g., <Go home> or <Follow leader>, which exhibit non-blocking behavior and execute concurrently with the invoker).

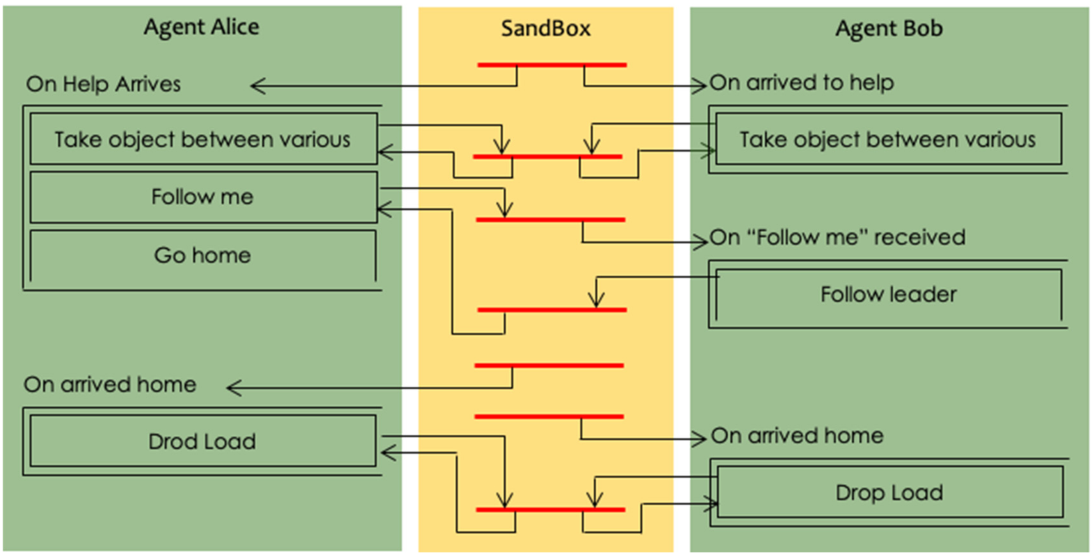


Figure 6. Multi-agent robotics primitives and synchronization in the distributed environment (red bars represent events transmitted from and to different agents).

It is noteworthy that the programming primitives for agents use terms and exhibit behaviors familiar to children, making them easy to use. When invoking these primitives, no parameters are required, as they are explicitly inferred from the context, which further simplifies their use, since children do not need to handle abstractions such as variables or parameters. In short, at this level, events are managed and actions are executed at a very high level of abstraction.

From an implementation perspective, the execution of behavior primitives is supported by a Python-based runtime that integrates real-time data from the computer vision module implemented with OpenCV. Robot states (position, orientation, and interaction events) are continuously updated and mapped into high-level symbolic events that trigger the corresponding primitives. This approach enables the abstraction of low-level perception and control into event-driven behaviors, allowing primitives to operate on semantically meaningful conditions (e.g., proximity, alignment, or group formation) without requiring explicit parameterization by the user.

3.6. High-Level Programming Interface (SandBox)

The SandBox layer provides high-level distributed programming primitives designed to support multi-robot collaboration scenarios. These primitives enable group formation and role assignment (e.g., leader election), spatial reasoning through logical arena partitioning, exploration and resource search strategies, collaborative manipulation behaviors, and coordination patterns such as help requests, leader following, or collective transport.

Primitives are intentionally designed using context-aware semantics that minimize the need for parameters or variable management. This design choice reduces cognitive load for novice learners and aligns with block-based programming paradigms emphasizing accessibility and expressiveness.

The platform also supports programming at lower abstraction levels (micro-robot control, communication, HAL), enabling progressive transitions from introductory activities to advanced distributed robotics experimentation.

These architectural components enable a hybrid learning environment in which distributed robotics concepts can be explored through embodied experimentation, immersive visualization, and progressive programming abstraction.

4. Validation

To assess the feasibility and educational usability of the proposed Spatial Augmented Reality multi-robot arena, we conducted a preliminary validation consisting of two complementary studies: (1) Technical validation of virtual sensing accuracy and responsiveness, including calibration and error analysis; (2) Exploratory user evaluation, examining the usability of the programming abstractions and learners' ability to implement collaborative multi-robot behaviors. All experiments were conducted using two Smart Cutebot robots (ElecFreaks, Shenzhen, China) equipped with BBC micro:bit V2 boards (Farnell, Leeds, UK) and fiducial markers enabling computer vision tracking.

We address the following research questions: (RQ1) What is the localization and orientation accuracy and end-to-end latency of the SAR-based virtual sensing pipeline across the workspace? (RQ2) To what extent can novice learners successfully implement a collaborative multi-robot task using the provided MakeCode abstractions with minimal instructor intervention? (RQ3) What are the practical constraints (lighting, occlusion, number of robots) affecting scalability and classroom deployment?

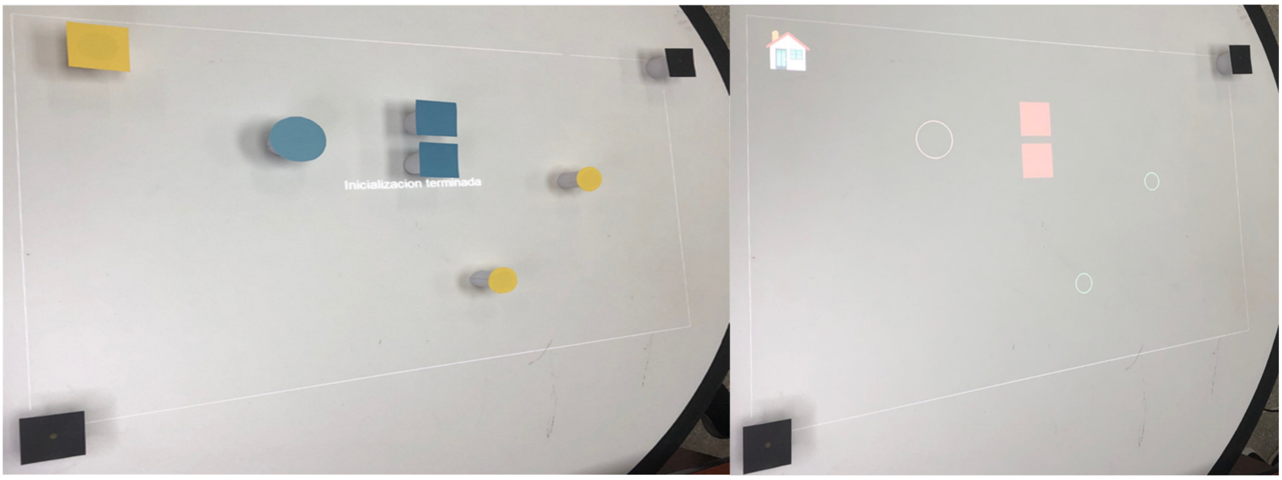
https://doi.org/10.3390/app16094145

4.1. Virtual Sensor Calibration and Performance Evaluation

Following some previous work that validated ideas related to virtual sensors and actuators [30,31], we conducted a set of experiments aimed to evaluate the accuracy, resolution, and response time of the virtual absolute position and orientation sensors relative to the arena coordinate system.

The arena workspace was first defined using fiducial markers establishing a global reference frame (see Figure 7). To improve measurement precision, calibration trials were performed using static markers placed at known coordinates rather than moving robots (see Figure 8).

Figure 7. Sandbox table with markers and details of the camera and projector.



(**a**) (**b**)

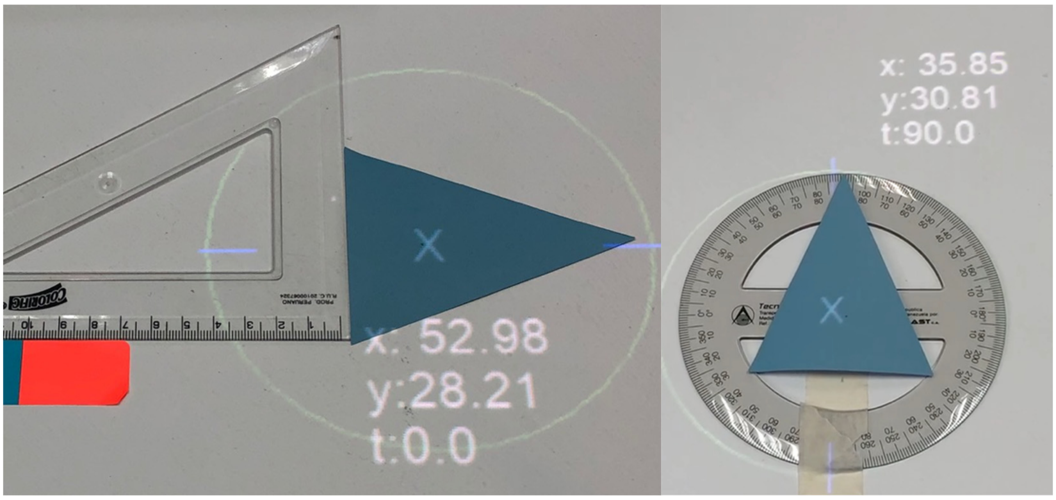
Figure 8. (**a**) Initialization of the agents using physical squares and circles; (**b**) Configuration of virtual elements based on the physical elements.

For each test point, the system estimated horizontal position (x), vertical position (y), orientation (theta). Ground-truth positions were obtained through manual measurement using a calibrated grid. Sensor estimates were recorded over multiple trials across the full workspace.

The sensor's performance was characterized by using response time to changes, range, resolution, Mean Absolute Error (MEA), Mean Absolute Percentage Error (MAPE): Mean Relative Error (MRE) $\times$ 100, accuracy: 100 $-$ MAPE, and precision ($\pm$SD). For characterization purposes, 15 samples were selected for each representative location; results are summarized in Table 1. Figure 9 shows the position and orientation evaluation process.

Table 1. Accuracy of the absolute position and orientation sensor.

	Range	Resolution	MAE	MRE $\times$ 100 $\pm$ SD	Accuracy	Response Time
Horizontal position	0–104 cm	0.1 cm	0.10 cm	0.32 $\pm$ 0.09	99.68%	0.02 s
Vertical position	0–62 cm	0.1 cm	0.09 cm	0.61 $\pm$ 0.05	99.39%	0.02 s
Orientation	0–359°	1.0°	0.58°	0.80 $\pm$ 0.48	99.20%	0.02 s



(**a**) (**b**)

Figure 9. Evaluation process of (**a**) position; (**b**) orientation.

These results indicate that the SAR-based virtual sensing infrastructure provides sufficiently accurate and responsive localization to support distributed coordination tasks in educational scenarios. Additional tests validated the operation of virtual collision detection sensors, which dynamically projected proximity zones around robots. Collisions with physical obstacles, virtual objects, or arena boundaries were detected reliably, complementing onboard optical sensors used as a safety mechanism to prevent robots from leaving the workspace.

4.2. Virtual Actuator Validation

To verify hybrid physical–virtual interaction capabilities, experiments were conducted using the virtual gripper actuator. Robots successfully performed these actions: (1) pickup of small virtual objects, (2) collaborative manipulation of large virtual resources, and (3) transport and placement within designated zones. These tests confirmed the correct synchronization between projected virtual entities, robot motion primitives, and event-driven coordination mechanisms. Figure 10 shows a sequence of the virtual gripper actuator's operation as it picks up, transports, and places an object near the designated area.

Figure 10. Agent (blue micro-robot) picking up, transporting, and placing a small object from home to the center of the arena.

4.3. User Study: Programming Distributed Multi-Robot Tasks

An exploratory usability study was conducted to evaluate whether novice learners could understand and apply the programming abstractions provided by the µBot and SandBox extensions.

Four children aged 8–11 with prior experience in block-based programming (Scratch) participated in the evaluation. A comparison group consisted of 6 undergraduate students enrolled in a robotics elective course (see Figure 11). The exploratory user study involved a small sample size and focused on short-term interaction; future work will include controlled classroom deployments and longitudinal learning assessment.

Figure 11. Students solving the collaborative transportation problem during the user study.

The study followed three stages. First was the Embodied algorithm design task. Participants were asked to collaboratively transport physical benches in a laboratory setting. They were then asked to verbalize their coordination strategies, effectively externaliz-

https://doi.org/10.3390/app16094145

ing distributed algorithms in natural language. Second was the Programming task. The same collaborative transport problem was implemented using SandBox programming blocks within the Microsoft MakeCode environment. Third was Execution and observation. Participants observed robots executing the programmed behaviors and iteratively refined their solutions. The evaluation focused on task completion success, number of instructor interventions required, qualitative usability feedback, observed understanding of coordination concepts.

4.4. Implementation Cost

The hardware costs associated with the prototype deployment are summarized in Table 2. The total system cost was approximately €1560. However, when existing classroom infrastructure such as computers and projectors is available, the incremental cost decreases to approximately €260, supporting the feasibility of classroom deployment.

Table 2. Costs associated with arena component.

Description	Quantity	Unit Price (€)	Total
Arena Surface Area	1	50	50
Projector Stand	1	80	80
Webcam	1	50	50
Projector	1	550	550
Computer	1	700	700
Cute:Bot Robot	2	35	70
Micro:bit Cards	3	20	60
			1560

5. Discussion

The results of the technical validation and exploratory user study provide initial evidence that the proposed Spatial Augmented Reality multi-robot platform can support both reliable hybrid physical–virtual interaction and accessible learning experiences focused on distributed robotics concepts.

The calibration experiments demonstrated that SAR-based virtual sensing can provide sufficiently accurate localization and collision detection to enable coordinated multi-robot behaviors in educational contexts. These findings are consistent with prior research on hybrid robotics environments and virtual sensing infrastructures, which emphasize the potential of external perception systems to extend robot capabilities without increasing hardware complexity.

This approach aligns with recent discussions in applied swarm robotics, which identify the reduction in hardware complexity and the use of external infrastructure as key enablers for scalable multi-robot systems [20]. In this sense, the proposed platform can be interpreted not only as an educational tool but also as a simplified example of emerging hybrid robotics paradigms.

Importantly, the ability to perform precise navigation and manipulation tasks using robots lacking onboard encoders or advanced sensors confirms the feasibility of using software-based sensing abstractions as pedagogical scaffolding mechanisms. This aligns with recent educational robotics research suggesting that virtual sensors can facilitate earlier engagement with complex coordination problems by reducing technical barriers associated with low-level control.

From a broader perspective, this contribution connects with the evolution of swarm robotics described by Cheraghi et al. [21], where the field is progressively transitioning from simulation-heavy approaches toward more accessible, embodied, and application-oriented systems. The proposed SAR environment contributes to this transition by lowering the barrier between conceptual understanding and physical experimentation.

The exploratory usability study indicates that learners with prior experience in block-based programming were able to understand and apply high-level coordination primitives to solve collaborative transport tasks. Participants rapidly adopted behaviors such as requesting assistance, following leaders, and jointly manipulating virtual objects. These observations suggest that the platform's abstraction mechanisms effectively support conceptual reasoning about distributed coordination, allowing learners to focus on strategy design rather than implementation details [9,32].

These findings are consistent with recent work in inclusive educational robotics, such as da Silva et al. [33], which shows that appropriately scaffolded robotics environments can support learners with diverse cognitive profiles, including those with learning difficulties. In this context, the use of high-level primitives and virtual sensing in our platform can be understood as a form of computational scaffolding that reduces cognitive load while preserving conceptual richness.

Furthermore, the strong engagement observed among younger participants, characterized by curiosity-driven interaction and experimentation, mirrors findings reported in recent systematic reviews on the use of augmented reality in robotics education [8]. However, this increased engagement must be interpreted cautiously. While immersive visualization can increase motivation and perceived learning value, the SAR setup also introduces a potential novelty effect, whereby learners' engagement may be driven primarily by the newness and technological appeal of the system rather than by the pedagogical approach itself. This concern is supported by findings in AR-based learning environments, where initial increases in motivation do not always translate into sustained learning gains once the novelty diminishes [27].

This issue is particularly relevant given the short duration of the interaction sessions and the fact that most participants were first-time users of the platform. Under these conditions, novelty effects can inflate perceived usability, enjoyment, and even self-reported learning. Consequently, the observed engagement should be interpreted as initial engagement rather than as evidence of sustained educational impact.

In contrast, university-level robotics students demonstrated a more analytical perspective, focusing on communication protocols, hardware constraints, and programming models. This divergence suggests that prior experience may partially mitigate novelty-driven responses, allowing more experienced learners to engage more critically with the system's abstractions, enabling the same platform to be used across educational stages.

The results suggest that integrating immersive visualization, hybrid sensing mechanisms, and progressive programming abstractions can help bridge the gap between introductory robotics activities and advanced distributed robotics topics. However, the potential influence of novelty implies that part of the platform's effectiveness may stem from its experiential and immersive qualities, raising the question of whether such engagement can be sustained over time and translated into deeper conceptual understanding.

By enabling learners to experiment with concurrent behaviors, message-based coordination, and stigmergic interaction patterns, the platform introduces computational models that differ from the sequential paradigms typically emphasized in early programming education. Exposure to such models may contribute to a broader understanding of computational thinking as the ability to select appropriate abstractions and coordination strategies for complex systems. Nevertheless, future research must determine whether these conceptual benefits persist once the novelty of the environment diminishes.

Several limitations must be considered when interpreting the results. First, the user study involved a small sample size and short interaction sessions, limiting the generalizability of findings regarding learning outcomes and long-term engagement. Second, the novelty associated with physical robots and immersive SAR projections may have significantly influenced participants' motivation, engagement, and perceived learning outcomes. Third, the validation focused on relatively simple collaborative tasks and did not evaluate performance in more complex swarm coordination scenarios or large-scale deployments. Fourth, the current prototype relies on a controlled laboratory setup with a single camera and projector, which may impose constraints on scalability, robustness under varying lighting conditions, and simultaneous interaction among larger groups of learners.

Future studies should therefore include controlled classroom experiments, explicitly accounting for novelty effects (e.g., through comparison groups using non-immersive interfaces or repeated exposure designs), longitudinal evaluations of learning progression, and scalability assessments involving larger robot teams and more diverse coordination challenges. Additionally, future work should explore the applicability of this approach in inclusive educational settings, building on recent evidence that educational robotics can support learners with diverse needs [33].

6. Conclusions

This paper presented a low-cost Spatial Augmented Reality arena designed to support programming activities involving multiple physical robots in educational contexts. The proposed platform integrates hybrid physical–virtual interaction, software-based virtual sensing and actuation mechanisms, and a layered programming architecture enabling progressive abstraction from micro-robot control to distributed multi-agent coordination.

Technical validation demonstrated that SAR-based virtual sensing provides accurate localization and responsive collision detection to support coordinated behaviors, even in robots lacking advanced onboard sensing capabilities. An exploratory user study showed that learners with prior experience in block-based programming were able to design and implement collaborative multi-robot solutions using high-level coordination primitives, suggesting the platform's potential to facilitate early exposure to distributed computational models.

By combining immersive visualization with tangible robotics interaction and accessible programming interfaces, the platform contributes toward addressing the fragmentation between introductory educational robotics tools and research-oriented multi-robot experimentation environments.

Future work will focus on conducting larger-scale classroom studies, evaluating learning outcomes across different age groups, and extending the platform with additional interaction modalities such as head-mounted augmented reality displays and richer multi-robot swarm scenarios. Planned developments also include the open-source release of software components, improved perception robustness, and the integration of adaptive learning analytics to support personalized learning trajectories.

Overall, the proposed SAR arena represents a step toward more integrated educational robotics ecosystems in which embodied interaction, immersive visualization, and distributed programming converge. By lowering technical barriers while preserving the richness of multi-robot experimentation, hybrid SAR environments may support new learning pathways connecting early computational thinking experiences with advanced cyber–physical systems education.

The main contribution of this work is demonstrating that projection-based augmented reality can function as a pedagogically meaningful coordination interface for embodied learning of distributed robotics concepts.

Author Contributions: Conceptualization, J.L.M., P.G. and A.E.; methodology, J.L.M., P.G. and A.E.; software, J.L.M.; validation, J.L.M.; writing—original draft preparation, J.L.M.; writing—review and editing, J.L.M., P.G. and A.E.; supervision, P.G. and A.E.; funding acquisition, J.L.M., P.G. and A.E. All authors have read and agreed to the published version of the manuscript.

Funding: This research was partially funded by DEUSTEK5 Research Group (IT1901-26).

Institutional Review Board Statement: The study was conducted in accordance with the Declaration of Helsinki and approved by the Institutional Review Board of the Universidad Católica Andrés Bello (UCAB) (CONSTANCIA 3-2026, 16 April 2026).

Informed Consent Statement: Informed consent was obtained from all subjects involved in the study.

Data Availability Statement: Data supporting reported results is available on request. Micro-robot and SandBox (v. 1.0) software is available on request.

Acknowledgments: During the preparation of this manuscript, the authors used OpenAI ChatGPT 5.4 Mini and DeepL Translate v. 26.14 for the purposes of translation to English and editing of the manuscript. The authors have reviewed and edited the output and take full responsibility for the content of this publication.

Conflicts of Interest: The authors declare no conflicts of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

Abbreviations

The following abbreviations are used in this manuscript:

SAR	Spatial Augmented Reality
STEM	Science, Technology, Engineering, and Mathematics
MRS	Multi-Robot Systems
MAR	Multi-Agent Robotics
SBR	Swarm-Based Robotics
VR	Virtual Reality
AR	Augmented Reality
MR	Mixed Reality
HAL	Hardware Abstraction Layer
MAE	Mean Absolute Error
MAPE	Mean Absolute Percentage Error
MRE	Mean Relative Error

References

1. Bonabeau, E.; Dorigo, M.; Theraulaz, G. *Swarm Intelligence—From Natural to Artificial Systems*; Oxford University Press: Oxford, UK, 1999.
2. Liu, J.; Wu, J. *Multiagent Robotic Systems*; CRC Press: Boca Raton, FL, USA, 2001.
3. Van Steen, M.; Tanenbaum, A.S. *Distributed Systems*; Maarten van Steen: Leiden, The Netherlands, 2017; Volume 3.
4. Aho, A. Computation and Computational Thinking. *Comput. J.* **2012**, *55*, 832–835. [CrossRef]
5. Broll, B.; Ledeczi, A.; Volgyesi, P.; Sallai, J.; Maroti, M.; Vanags, C. Introducing parallel and distributed computing to K12. In *Proceedings of the 2017 IEEE International Parallel and Distributed Processing Symposium Workshops*; IEEE: Piscataway, NJ, USA, 2017; pp. 323–330.
6. Papert, S. *Mindstorms: Children, Computers, and Powerful Ideas*; Basic Books, Inc.: New York, NY, USA, 1980.
7. Resnick, M.; Maloney, J.; Monroy-Hernández, A.; Rusk, N.; Eastmond, E.; Brennan, K.; Millner, A.; Rosenbaum, E.; Silver, J.; Silverman, B.; et al. Scratch: Programming for all. *Commun. ACM* **2009**, *52*, 60–67. [CrossRef]
8. Prokopiuk, N.; Falkowski, P. Applicability of augmented and virtual reality for education in robotics. *J. Autom. Mob. Robot. Intell. Syst.* **2023**, *16*, 65–74. [CrossRef]

9. Yang, F.C.O.; Lai, H.M.; Wang, Y.W. Effect of augmented reality-based virtual educational robotics on programming students' enjoyment of learning, computational thinking skills, and academic achievement. *Comput. Educ.* **2023**, *195*, 104721. [CrossRef]
10. Bimber, O.; Raskar, R. *Spatial Augmented Reality Merging Real and Virtual Worlds*; A K Peters: Wellesley, MA, USA, 2005.
11. Ghiringhelli, F.; Guzzi, J.; Caro, G.; Caglioti, V.; Gambardella, L.; Giusti, A. Interactive Augmented Reality for understanding and analyzing multi-robot systems. In *Proceedings of the 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*; IEEE: Piscataway, NJ, USA, 2014; pp. 1195–1201.
12. Hönig, W.; Milanes, C.; Scaria, L.; Phan, T.; Bolas, M.; Ayanian, N. Mixed reality for robotics. In *Proceedings of the 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*; IEEE: Piscataway, NJ, USA, 2015; pp. 5382–5387.
13. Chen, M.; Zhang, P.; Wu, Z.; Chen, X. A multichannel human-swarm robot interaction system in augmented reality. *Virtual Real. Intell. Hardw.* **2020**, *2*, 518–533. [CrossRef]
14. Hénard, A.; Peillard, É.; Rivière, J.; Kubicki, S.; Coppin, G. Towards Augmented Reality Support for Swarm Monitoring: Evaluating Visual Cues to Prevent Fragmentation. *IEEE Trans. Vis. Comput. Graph.* **2025**, *31*, 10142–10151. [CrossRef]
15. 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 2022 CHI Conference on Human Factors in Computing Systems*; ACM: New York, NY, USA, 2022; pp. 1–33.
16. Benitti, F.B.V. Exploring the educational potential of robotics in schools: A systematic review. *Comput. Educ.* **2012**, *58*, 978–988. [CrossRef]
17. Sanusi, I.T.; Oyelere, S.S.; Vartiainen, H.; Suhonen, J.; Tukiainen, M. A systematic review of teaching and learning machine learning in K-12 education. *Educ. Inf. Technol.* **2023**, *28*, 5967–5997. [CrossRef]
18. Toh, L.; Causo, A.; Tzuo, P.-W.; Chen, I.-M.; Yeo, S. A Review on the Use of Robots in Education and Young Children. *Educ. Technol. Soc.* **2016**, *19*, 148–163.
19. Bers, M. Coding and Computational Thinking in Early Childhood: The Impact of ScratchJr in Europe. *Eur. J. STEM Educ.* **2018**, *3*, 8. [CrossRef]
20. Kegeleirs, M.; Birattari, M. Towards applied swarm robotics: Current limitations and enablers. *Front. Robot. AI* **2025**, *12*, 1607978. [CrossRef]
21. Cheraghi, A.R.; Shahzad, S.; Graffi, K. Past, Present, and Future of Swarm Robotics. In *Intelligent Systems and Applications. IntelliSys 2021*; Arai, K., Ed.; Lecture Notes in Networks and Systems; Springer: Cham, Switzerland, 2022; Volume 296. [CrossRef]
22. Giraldo, C.; Florian-Gaviria, B.; Bacca-Cortes, E.; Gómez, F.; Muñoz, F. A programming environment having three levels of complexity for mobile robotics. *Ing. Investig.* **2012**, *32*, 76–82. [CrossRef]
23. Dede, C. Immersive Interfaces for Engagement and Learning. *Science* **2009**, *323*, 66–69. [CrossRef]
24. Radianti, J.; Majchrzak, T.A.; Fromm, J.; Wohlgenannt, I. A systematic review of immersive virtual reality applications for higher education: Design elements, lessons learned, and research agenda. *Comput. Educ.* **2020**, *147*, 103778. [CrossRef]
25. Kapinus, M.; Beran, V.; Materna, Z.; Bambušek, D. Augmented reality spatial programming paradigm applied to end-user robot programming. *Robot. Comput.-Integr. Manuf.* **2024**, *89*, 102770. [CrossRef]
26. Walker, M.; Phung, T.; Chakraborti, T.; Williams, T.; Szafir, D. Virtual, Augmented, and Mixed Reality for Human-robot Interaction: A Survey and Virtual Design Element Taxonomy. *ACM Trans. Hum.-Robot Interact.* **2023**, *12*, 1–39. [CrossRef]
27. Makhataeva, Z.; Varol, H.A. Augmented reality for robotics: A review. *Robotics* **2020**, *9*, 21. [CrossRef]
28. Pasalidou, C.; Lytridis, C.; Tsinakos, A.; Fachantidis, N. Augmented reality and robotics in education: A systematic literature review. *Comput. Hum. Behav. Artif. Hum.* **2025**, *4*, 100157. [CrossRef]
29. Jang, S.; Yum, H.; Lee, A. Data-Driven Forward Kinematics for Robotic Spatial Augmented Reality: A Deep Learning Framework Using LSTM and Attention. *Actuators* **2025**, *14*, 569. [CrossRef]
30. Lárez, J. *Ambiente de Robótica Multi-Agente Basado en Sensores y Actuadores Virtuales para Micro-Robots Minimalistas*; Universidad Católica Andrés Bello: Ciudad Guayana, Venezuela, 2020.
31. Espejo, P. Entorno de Robótica Educativa Multiagente Orientado a Favorecer el Desarrollo del Pensamiento Computacional en Jóvenes Cursantes de Educación Media. Bachelor's Thesis, UCAB, Escuela Ingeniería Informática, Puerto Ordaz, Venezuela, 2022.
32. Xu, F.; You, H.; Zhou, T.; Du, J. Augmented Reality Robot Manipulation with Physical Mock-up and Improved Spatial Cognition. *Int. J. Hum.-Comput. Stud.* **2025**, *205*, 103658. [CrossRef]
33. da Silva, M.D.G.T.; Araújo Maia, C.J.D.N.; Fernandes Curvelo, C.D.C.; Garcia, L.T.D.S.; Gonçalves, L.M.G. Educational robotics as a pedagogical resource for K-12 students with learning difficulties. *Sci. Rep.* **2025**, *15*, 35923. [CrossRef] [PubMed]