Internship report M2 SIIA

Immersive and localized data visualization of a robot swarm in Augmented Reality

Nolwenn Paluet

Internship carried out from 01/03/2023 to 31/08/2023

Internship tutors: Dr. PEILLARD Étienne (etienne.peillard@imt-atlantique.fr)
Mr. HÉNARD Aymeric (aymeric.henard@univ-brest.fr)

Referent teacher: Pr. DUVAL Laurence (laurence.duval@univ-brest.fr)

Éducational institution: Université de Bretagne Occidentale / Faculté des Sciences et Techniques
Internship host company: IMT Atlantique, 655 Av. du Technopôle 29280 Plouzané

Acknowledgments

I would like to thank Étienne PEILLARD and Aymeric HÉNARD, my supervisors at IMT Atlantique, for allowing me to carry out this internship and supporting me throughout.

I would like to thank the INUIT team of the Lab-STICC and the IT department of IMT Atlantique for their warm welcome and the professional experience they have given me.

Finally, I would like to express my gratitude to the interns at CERV and IMT-Atlantique, with whom I've become friends and shared some great moments during my internship.

Abstract

I did my final internship for the Master in Intelligent, Interactive and Autonomous Systems in the INUIT team at the Interaction division of Lab-STICC. My internship topic was Immersive and Localized Data Visualization in Augmented Reality and contributed to the ARTUISIS project, Augmented Reality and Tangible User Interface to Supervise and Interact with Robot Swarms. The goal was to create an Augmented Reality application to visualize localized data from a swarm of robots in order to understand and apprehend their behavior. I started by setting up communication between the AR headset and the gateway (a software program that allows all components of the environment to communicate) before calibrating the virtual environment in which my visualizations would be instantiated. I then implemented some simple data visualizations, such as the robots' direction vector and the swarm's convex envelope. This application can be used in a simulated environment or in a motion capture system with real robots. Implementing more relevant visualizations, such as observing the force that dominates a robot's behavior, and conducting a user study to determine their relevance are future tasks.

Key words: Augmented Reality, Situated visualizations, Robot swarm

Abstract

J'ai réalisé mon stage de fin d'étude du master Systèmes Intelligents, Interactifs et Autonomes au sein de l'équipe INUIT du pôle Interaction du Lab-STICC. Mon sujet de stage était la visualisation immersive et localisée de données en Réalité Augmentée et contribue au projet ARTUISIS, Augmented Reality and Tangible User Interface to Supervise and Interact with robot Swarms. Le but était de réaliser une application en Réalité Augmentée pour visualiser les données localisées d'un essaim de robots afin de comprendre et d'appréhender leurs comportements. J'ai d'abord mis en place la communication entre le casque de RA et la Passerelle (un logiciel permettant de faire communiquer tous les composants de l'environnement) avant de calibrer l'environnement virtuel où sont instanciées mes visualisations. J'ai ensuite implémenté quelques visualisations de données simples comme le vecteur de direction des robots ou encore l'enveloppe convexe de l'essaim. Cette application peut être utilisée dans un environnement simulé ou dans un système de capture de mouvement avec de vrais robots. Implémenter des visualisations plus pertinentes, comme observer la force qui domine le comportement d'un robot, et réaliser une étude utilisateurs pour déterminer la pertinence de celles-ci sont les futurs travaux à réaliser.

Mots clés: Réalité Augmentée, Visualisations localisées, Essaim de robots

Contents

1 Introduction

Currently in my final year of the Intelligent, Interactive and Autonomous Systems Master's degree, I am doing my end-of-studies internship in a research laboratory to validate my master's degree.

I chose to do my internship in a research laboratory to allow me to have more experience in computer science research as I want to do a Ph.D. after my Master's degree.

This internship was proposed by Etienne Peillard, an associate professor at IMT-Atlantique, within Lab-STICC, a research laboratory. The internship lasted from the 1st of March 2023 to the 31st of August 2023 at IMT-Atlantique, in Brest. The project focused on immersive and localized data visualization of a robot swarm in Augmented Reality. The remainder of this report is organized as follows. First, I will give the context of the internship, such as the host company, the project, and the work environment. Second, I will present my work done during the internship. Then, I will evoke the difficulties encountered and how I dealt with them, before talking about the work that remains to be done. I will finish by giving my personal assessment and a conclusion about my work.

2 Context

2.1 Host company

L'École nationale supérieure Mines-Télécom Atlantique Bretagne Pays de la Loire, also known as IMT-Atlantique, is a renowned institution that holds a prominent position in the field of higher education and research. This institution was created in 2017 from the merger of École Nationale Supérieure des Mines de Nantes and Télécom Bretagne and is headed by Christophe Lerouge. The aim of the institution is to train the next generation of engineers and professionals in various fields such as digital, energy and environment. The institution is located on three campuses: Brest, Rennes and Nantes.

2.2 Research Laboratory

This internship took place within the Lab-STICC (Laboratoire des Sciences et Techniques de l'Information, de la Communication et de la Connaissance) in the INUIT team of the Interaction pole. This laboratory is divided into 9 research divisions and 25 research teams. Each division deals with a specific subject.

The Interaction pole is a multi-disciplinary division whose aim is to improve interactions between users and systems. This pole is divided into 3 teams: the COMMEDIA team aims to improve system engagement, the RAMBO team aims to make them as autonomous as possible and the INUIT team aims to improve their naturalness. The INUIT (Immersive Natural User Interaction Team) works on various themes such as Augmented, Virtual and Mixed Reality, 3D and tangible interactions, and complex multi-user systems.

2.3 Project

This internship is a part of a thesis conducted by Aymeric Hénard called ARTUISIS[1], Augmented Reality and Tangible User Interface to Supervise and Interact with robot Swarms. The aim of the ARTUISIS project is to help an operator understand the behavior of a swarm of robots using augmented reality and to influence it through a tangible interface (see Fig. 1).

A robot swarm is a decentralized system of autonomous robots that possesses self-organizing properties, which can lead to emergent behavior. These emerging behaviors make it difficult for an operator to supervise and interact with a robot swarm.

My internship contributes to the augmented reality part of the project, i.e. the supervision of a robot swarm by visualizing their data in an immersive and localized

[1] https://siia.univ-brest.fr/artuisis/

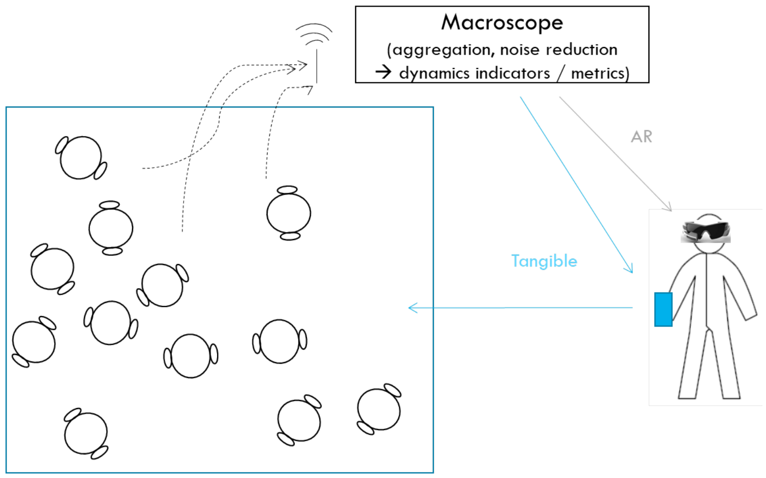


Fig. 1: ARTUISIS platform representation

way. The goal is to display numerical information, like swarm health, robot speed, or interactions between agents, in a spatialized manner. The data can be related to the environment, the swarm or the user. The visualizations must therefore be adapted to make them as understandable as possible.

To achieve this, we first need to define the visualizations to be tested and then, implement and test them on the ARTUISIS project platform, which links robots, AR and the tangible interface.

2.4 Project management

During my internship, I was mentored by the researchers of the INUIT team Étienne Peillard and Aymeric Hénard. I was invited to team meetings where researchers shared their work up to twice a month. I also had the opportunity to listen to Ph.D. students practicing for their CSI (Comité de Suivi Individuel), which gave me an insight into thesis follow-up.

About my internship itself, I was working by myself on the augmented reality part of the project but I sometimes had help from several people during the various development phases.

To supervise the work done and show my progress during the internship, I had weekly meetings with Etienne Peillard, my supervisor. At the beginning of the internship, I was asked to make a weekly work schedule for the first 3 months. I think that this kind of planner is a good way to go if you know how to manage your time and plan realistic tasks. That was not the case for me and I decided to do things differently.

I also set up a Notion[2] environment, a note-taking application, at the beginning of

[2] https://www.notion.so/fr-fr

the internship to take daily notes and to have a visual representation of the progress of my work, but I struggled a lot to keep it up to date.

Each week, I had to send an email to my supervisor writing down the week's progress, the problems I had encountered and what I had planned for the following week. These emails were very useful to keep track of the work done, and sending them helped me to be more assiduous. In addition to these emails, I had weekly meetings where I could show my application, ask questions and solve problems.

2.5 Work environment

Throughout my work, I had to use a variety of software and hardware tools.

2.5.1 Hardware

I had the opportunity to use different kinds of equipment, from an AR headset to a motion capture system.

- Magic Leap

  The AR headset I used is the Magic Leap 2, released in September 2022. This device has one of the widest fields of view (70°) on the market and it is easy to use with the Unity game engine. The headset is used with a controller, which is more precise than hand control. This AR headset can be used with the MagicLeap Hub software. It allows a user to manage the files, update the device, simulate and stream it.

- MONA robots

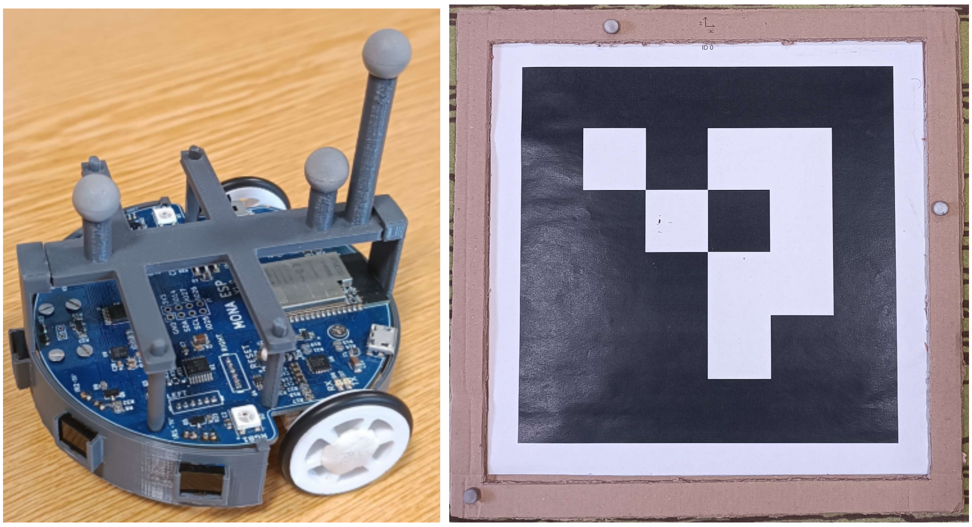
  The robot used in the swarm is the MONA. MONA is a 2-wheeled small robot (80 mm in diameter) entirely open-source and customizable. It is equipped with sensors and actuators such as motors, IR receivers and transmitters, and LED (see Fig. 2(a)). It can be programmed using the Arduino IDE.

- OptiTrack

  OptiTrack is a motion capture system that uses multiple cameras to track robots in real-time, in my internship. This platform allows us to get precise robot positions and rotations. The robots are tracked using sensors and each robot has its own constellation (a set of three sensors) to be distinguished. The Optitrack system has to be calibrated to have good tracking of each object.

- Marker

  The motion capture system does not track the AR headset. To be able to calibrate the headset environment with the OptiTrack environment, I use a marker tracked



(a) (b)

Fig. 2: (a) MONA robot (b) Marker

by both the AR headset and the OptiTrack (see section 3.2.2). The marker is an ArUco[3], a square marker with a wide black border and an inner binary matrix. More precisely, I use a 4x4 marker. To be able to track it, I made a cardboard frame to hold the three motion capture sensors (see Fig. 2(b)).

2.5.2 Software

- Unity (C#), VScode and Git

  The augmented reality application was developed using Unity, a multiplatform game engine developed by Unity Technologies. To write the source code, I used Visual Studio Code, a code editor and to manage my project I used Git, a decentralized version management software. I already used these different tools during my classes and using them during my internship enabled me to learn how to use them better.

- Gateway

  The gateway is a software, developed in Python, which connects the various components of the ARTUISIS platform. This software can be used in two different modes: a simulation mode and an OptiTrack mode. In simulation mode, it connects Webots, i.e. robots and a supervisor, and the AR headset for data exchange. In order to do so, each component has its own IP address simulated by a network card using a virtual machine (see Fig. 3).

[3] https://chev.me/arucogen/

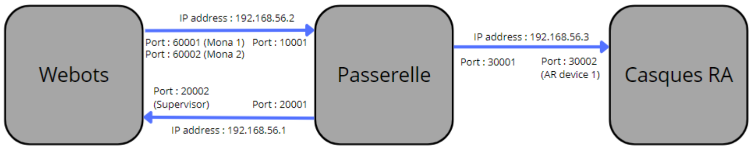


Fig. 3: Component connections in simulation mode

In Optitrack mode, we use a router to manage the network. All the hardware components are connected via the router and the gateway exchanges data between the OptiTrack, the robots and the AR headset (see Fig. 4).

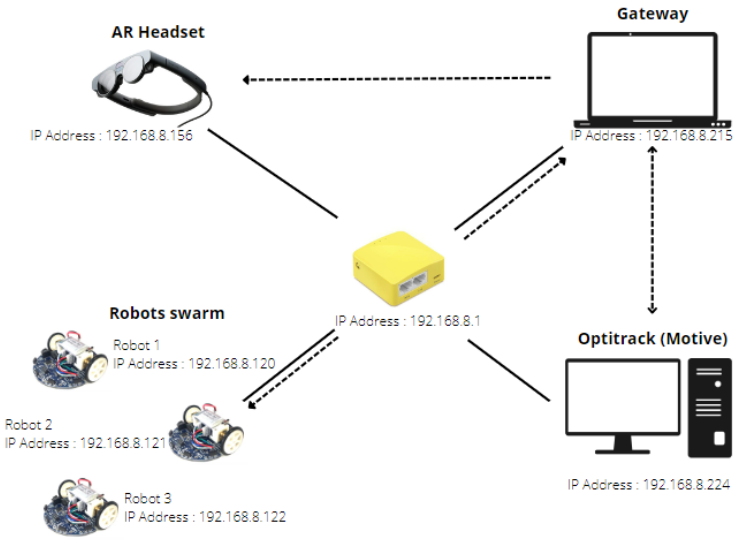


Fig. 4: Component connections in OptiTrack mode

- Webots

  For the simulation part of my work, I used Webots, an open-source robot simulation software. Developed by Cyberbotics Ltd, it is widely used in industry, research and education. Physics and inertia are simulated using the Open Dynamic Engine (ODE), an open-source library for collision simulation and rigid body dynamics. I have already used this software in previous works so it was not difficult to use it.

3 Contribution

3.1 Related Works

3.1.1 Localized data representation

Before starting my internship, I carried out a literature review on techniques for localized data representation in Augmented Reality. Augmented Reality allows a user to overlay data over the real-world environment with different tools such as AR headsets or mobile devices. Generally speaking, visualizations are no different depending on the environment in which they are viewed as they are not linked to their environment. In the case of localized data visualization, the environment provides a context for understanding the data and these visualizations are called "Context-driven visualizations"[1]. There are several types of context-driven visualization, such as situated visualization, object as context, sensors as context and scene as context. Another way to display data that is strongly related to situated visualization is embedded visualization[2]. All these visualizations rely on the physical world, and more precisely, the physical presentation of the data is linked to a physical referent (see Fig. 5).

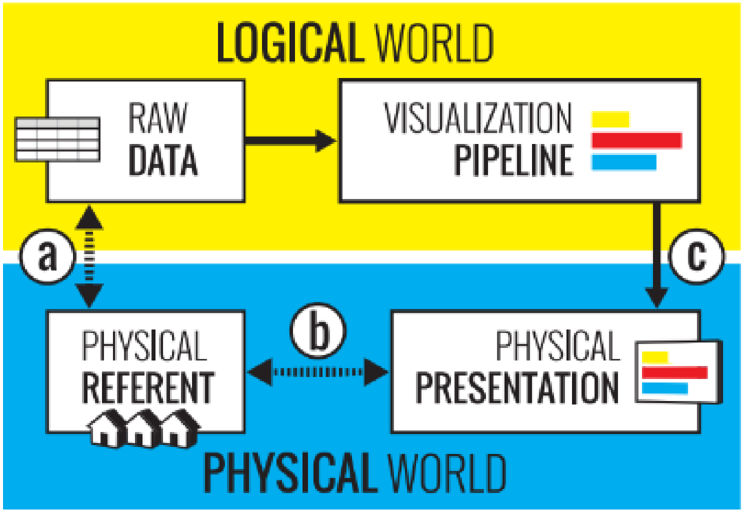


Fig. 5: A traditional visualization pipeline extended to the physical world. Raw data and data presentations are both linked to a physical referent. From Willett et al., (2017) [2].

A situated visualization is based on one physical presentation and it has to be "located close to the data's physical referent"[2]. The physical presentation can be spatially, temporally or geographically close to the physical referent. An embedded visualization uses several physical presentations, one for each physical referent. That means that each visualization can be displayed alone and each of them is tied to a real-world element with a low spatial indirection (see Fig. 6). Object, Sensor data and Scene as context use a specific focus to display related data[1]. For example, you may want to display a robot's characteristics or visualize data from one of its sensors.

In the context of my internship, robot data visualizations need to be superimposed. The visualizations implemented must therefore be embedded and they also use an object as context and sensor data as context.

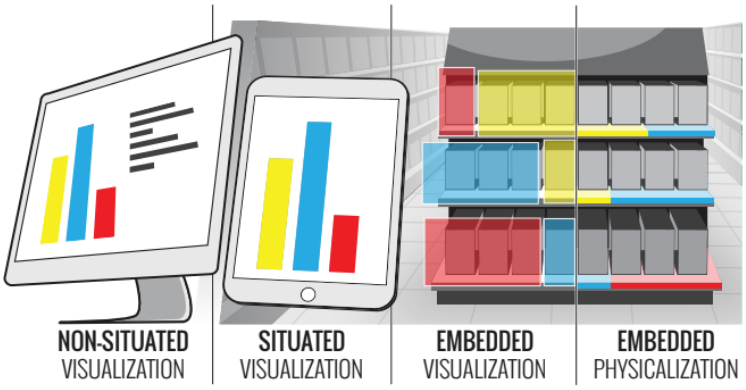


Fig. 6: Example of a situated and embedded visualization. From Willett et al., (2017) [2].

3.1.2 AR data visualizations for robot swarms

At the beginning of my internship, I looked up a few articles on augmented reality data visualizations applied to a swarm of robots. Robot data can be represented and displayed using one or more visualization techniques. Each visualization can be altered depending on the value of the data. An example of that would be that an arrow showing the direction of a robot could be varying in length in order to represent the robot's speed. The link between the robots can be color-coded depending on the relationship between the robots. These visualizations can be spatially situated.

The work of Ghiringhelli et al. (2014)[3] shows different kinds of visualization techniques applied to a robot swarm. They highlighted 3 types of data to be displayed: Textual information, which refers to debug messages, Symbolic information, which refers to graphical representation of the internal state of a robot and the last one is Spatially-situated data which refers to data that is related to a robot and has a location. They display data over a live video, spatially-situated means that the data is related to a robot, if the robot moves, the visualization will move with it whereas textual and symbolic information is always displayed next to the robot so it will not move if the robot moves.

Millard et al. (2018)[4] work is based on Ghiringhelli et al. (2014)[3] but the authors want to make the identification, localization and debugging easier. They get the information from a live video and the robots. The data displayed over the video comes from the robots. Data can be visualized using real-time charts like bar charts, pie charts, or line graphs instead of text. It can be used to represent how long a robot has been in any given state. A singular robot can be selected to visualize its data and the user can choose which data they want to show.

The work of Wheeler et al. (2019)[5] shows another work where data is displayed on top of a video. They visualize information using bar charts, pie charts, and line graphs. The data can be numerical, textual, or boolean. Pie charts can be used to represent data from a robot or the swarm and bar charts can be used to represent the data from a robot's sensors. The visualization is displayed above the robots. Libby, Zhong, et al. (2020)[6] have taken this work and adapted it for use with an augmented reality headset. The data displayed relates to the current state of a robot (position, rotation), the intended action (sensors data from all robots combined), or the sensors and actuators of a robot. The data is displayed using labels, graphs, and meshes.

In the end, the visualizations used are fairly simple. A lot of work uses pie and bar charts as well as line graphs and the displayed data mainly concerns the state of a robot. In these works, the visualizations implemented may not be localized and are displayed over a video. In the context of my internship, I am going to use an AR headset to view data from robots in real-time. In order to visualize the robot data, I will use labels, graphs and meshes, as shown in the work of Libby, Zhong, et al. (2020)[6].

3.2 Realisation

My work was divided into three parts. First, I have adapted the gateway code to enable data exchange between it and the AR headset. Then, I calibrated the AR headset in the environment and finally, I implemented the various visualizations. For the network and calibration parts, the work has been carried out in both simulation mode and OptiTrack mode. The simulation mode uses Webots to simulate the robots and can be activated in any environment. The OptiTrack mode uses real robots and requires a motion capture system.

3.2.1 Network

The first step of development was to establish data exchange between the AR headset and the gateway. The gateway was already programmed for data exchange between the gateway and Webots or the OptiTrack so I decided to use the same code to program data transmission from the gateway to the AR headset. The communication protocol used is UDP, the server is coded in Python while the client is coded in C#. The UDP protocol allows connectionless communication, but here I will use the term 'connection' to refer to the fact that the IP address and port entered by a user are correct and that there is an exchange of data between the client and the server. Communications are the same whether in simulation mode or OptiTrack mode.

- Server
  On the server side, once it has been activated we can first connect each robot by switching them on and then connect the AR headsets (see Fig. 7). Each robot sends a connection message and the gateway saves them in a list, it is the same principle for AR headsets. Once all components are connected, the gateway retrieves robots' data from Webots or the OptiTrack, updates the message sent for each robot, and sends it. For the AR headset, the gateway sends one message per robot which contains several data. In OptiTrack mode, the gateway also retrieves data from the marker and sends it to the AR headset.

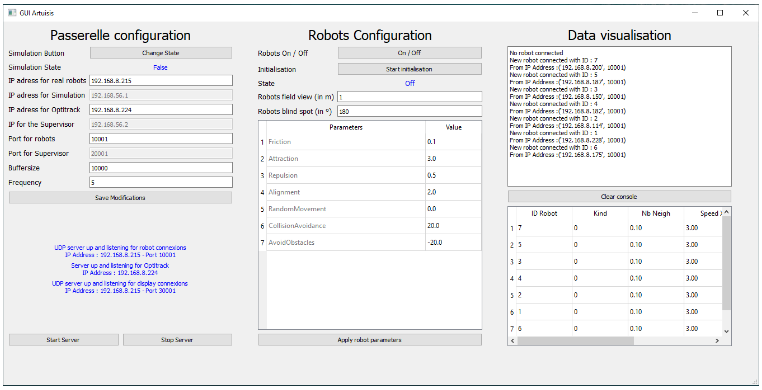


Fig. 7: Gateway interface

- Client
  On the client side, it uses two sockets. One is for sending the connection message to the server and the other is for receiving data (see Fig. 8). The user can enter the server's IP address and two ports. The AR headset will then send a message with its ID and the port for receiving messages to connect to the server. When the client receives messages, it checks the ID received and updates the object data corresponding to it.

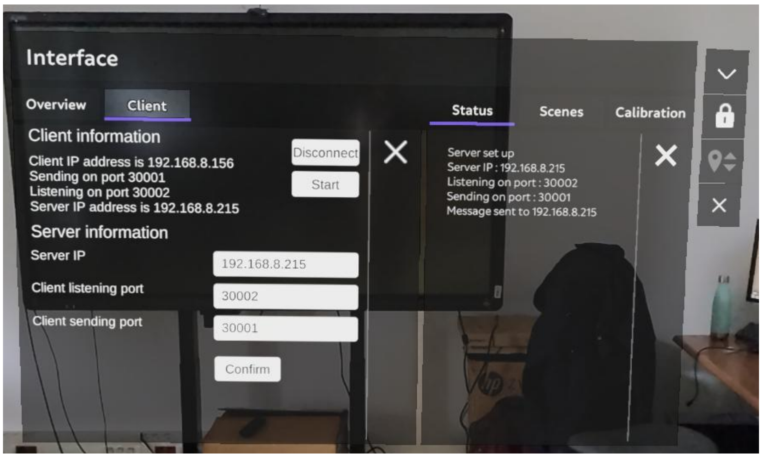


Fig. 8: Client interface

3.2.2 Calibration

Calibration means ensuring that the different environments have the same common frame of reference, to do so, I used a marker that is tracked in both environments (see section 2.5.1). The different environments are the AR headset environment, which has its origins in the XR Rig, and the OptiTrack environment.

To ensure that visualizations are aligned with robots, I decided to create an empty GameObject in the Unity scene where the various GameObjects of the visualizations will be instantiated and it is this object that will be moved to the correct position and rotation. To simplify, I move the AR headset environment to the Optitrack environment (see Fig. 9).

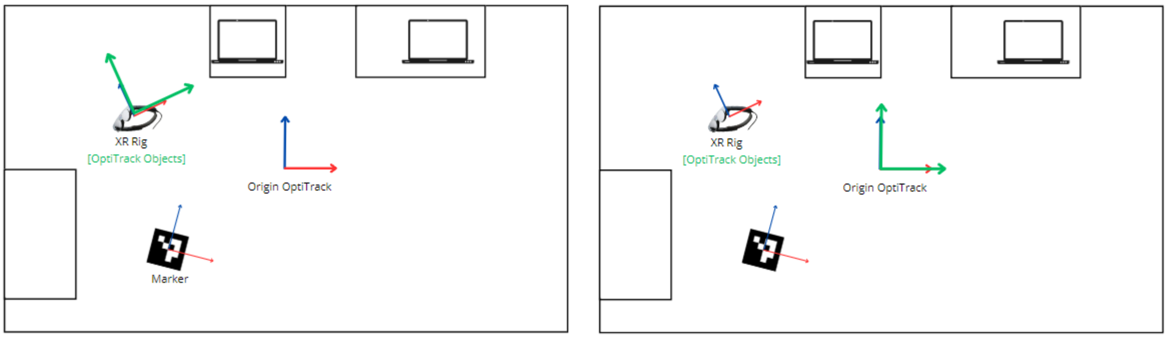


Fig. 9: Calibration representation in the environment

To align the two environments, I start by calculating the angle of rotation between my virtual environment and that of the OptiTrack. To do this, I calculate the difference in rotation between the XR Rig and the marker being tracked by the AR headset. Then I subtract the rotation of the real environment marker from this difference. Finally, I apply the rotation to my virtual environment. It is the same process for the position, I add the position of the XR Rig to that of the marker seen by the AR headset, then subtract the position of the marker from the real environment. Next, I apply the translation to my virtual environment.

```
// -- Rotation -- //
Vector3 rotDiff = Vector3.zero;

// Rotation difference between the XR Rig and the marker in Unity
rotDiff.y = xrOrigin.rotation.eulerAngles.y + markerOrigin.transform.rotation.eulerAngles.y;
// Sub the real Rotation of MarkerOrigin
rotDiff.y -= realRotation.eulerAngles.y;

optiTrackEnvironment.Rotate(rotDiff, Space.Self);

// -- Position -- //
Vector3 posDiff = xrOrigin.position + markerOrigin.transform.position;

// Position of the OptiTrack (in Unity) + difference between the markerOrigin and the XR Rig - Offset
optiTrackEnvironment.position = optiTrackEnvironment.position + posDiff - realPosition;
```

Although the calculation remains the same, the calibration procedure differs between the simulation mode and OptiTrack mode.

- Simulation
  In simulation mode, the calibration is a one-step process. It is useful for testing purposes, as visualizations do not need to be aligned with real robots. The virtual environment is initialized where the marker is located in the real environment, i.e. the actual position and rotation of the marker in the real environment is considered to be (0, 0, 0). When the AR headset recognizes the marker, the user can choose whether or not to calibrate.

- OptiTrack
  In OptiTrack mode, the calibration is a two-step process. The first step is similar to the simulation mode and is used to initialize the OptiTrack origin in the virtual environment. The position and rotation of the marker are those received by the client from the server and correspond to the position and rotation of the marker in the motion capture environment. This step provides an approximation of the location of the visualizations, but they will be offset. The second step is to correct this offset by moving the marker in the environment and taking a second measurement of its position, using the AR headset and OptiTrack.

This second measurement is used to calculate the angle between the virtual environment and the marker seen through the headset, and the virtual environment and the marker seen through the OptiTrack. I then rotate my virtual environment from this angle (see Fig. 10 and 11).

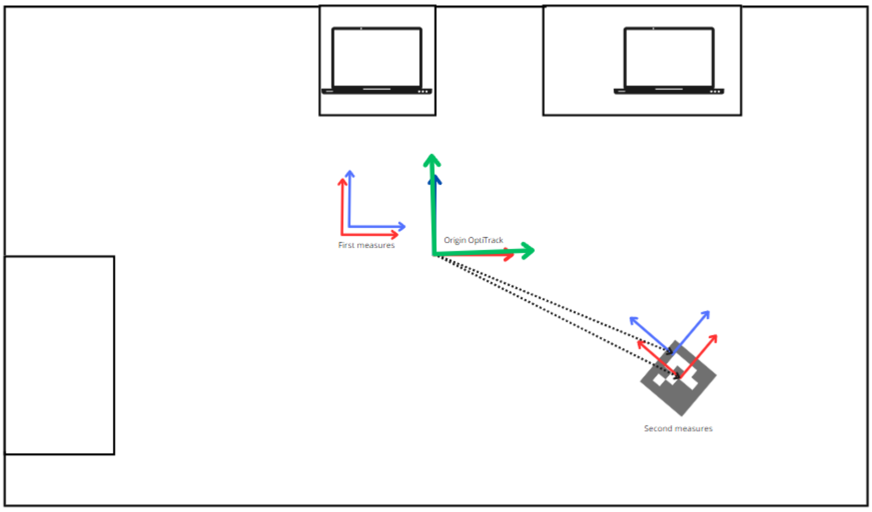


Fig. 10: Offset measurement

(a) (b)

Fig. 11: (a) Before calibration (b) After calibration

3.2.3 Visualizations

Once the client is connected and the environment calibrated, the user can display several visualizations[4] at once or individually. Several of these visualizations highlight or relate to clusters or fractures. A cluster in swarm robotics refers to a group of robots that are physically close to each other and often exhibit some degree of coordinated behavior or interaction. A fracture refers to a situation where the cohesive and coordinated behavior of a swarm becomes disrupted or broken. The visualizations implemented come from various scientific articles or ideas proposed by Aymeric.

- Directional vector

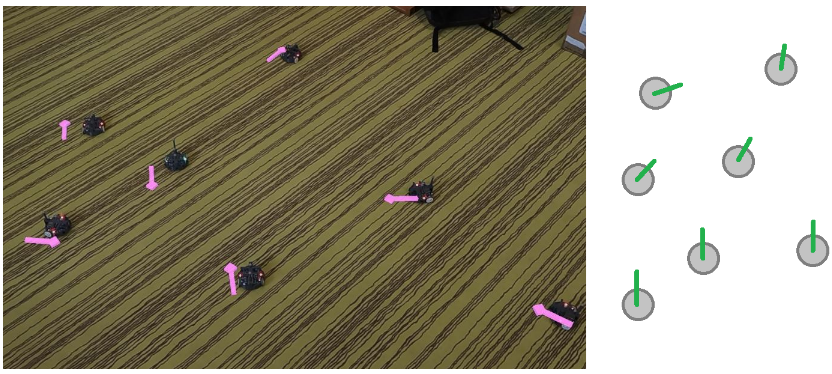


Fig. 12: Directional vector

  Indicates the direction of a robot with an arrow [3, 6, 7, 8, 9] (see Fig. 12). I chose to use an arrow to visualize direction because it was the most logical thing that came to mind. This visualization is applied to the swarm but could be used with selected robots or groups of robots. This visualization is not necessary as a human is able to notice this information. However, it can be improved as this visualization could also show the speed of a robot by adjusting the length of the arrow depending on its speed. Some other ways would be to show the internal movements of the swarm [10] or the overall direction [7, 11].

- Links between agents

  Allows a user to see links between robots, i.e. agents that perceive each other, neighbors, and groups. This visualization becomes unreadable when all robots in the swarm perceive each other and are neighbors [11, 12] (see Fig. 13). The implemented visualization shows the strength of the link. Depending on the distance between the robots, the link is more or less strong. The strength of the link could also be represented using colors or dotted lines. In AR, the use of environment space can also be interesting, as the strength of the link can be represented by its height. This visualization is applied to the swarm, but

[4] The screenshots show an offset that is not present in reality.

Fig. 13: Links between agents

selecting a robot could show the user all of its neighbors.

- At-risk Links between agents

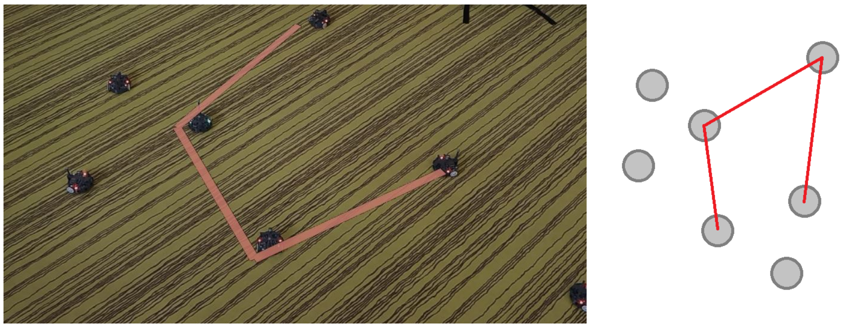


Fig. 14: At-risk links between agents

Allows a user to view inter-group links, i.e. links corresponding to the closest robots between two clusters. To find these robots, we build the minimum spanning tree of the swarm and retrieve its largest branches, i.e. the most distant robots. The farthest agents in the spanning tree correspond to the closest agents between clusters (see Fig. 14). The link is represented by a line but could be emphasized by flashing it to indicate to the user that this is something to monitor.

- Convex envelope

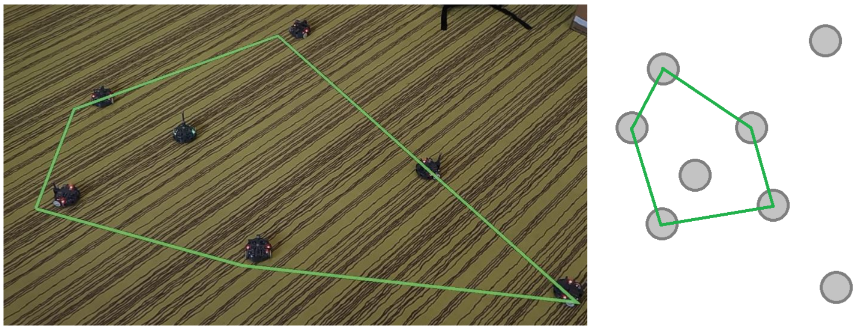


Fig. 15: Convex envelope

Allows the user to visualize the convex envelope of robot groups [7, 11, 13]. This visualization does not overload the user with data and can be used with other visualizations (see Fig. 15). The envelope is constructed by connecting the center of each robot with a line but this can lead to misunderstandings, such as thinking that the robots in the center are not part of the envelope. A way to improve this could be to place the robots within the lines or color the area. To take advantage of the height, it could also be represented as a dome, leaving room for other visualizations instead of coloring the area.

- Community

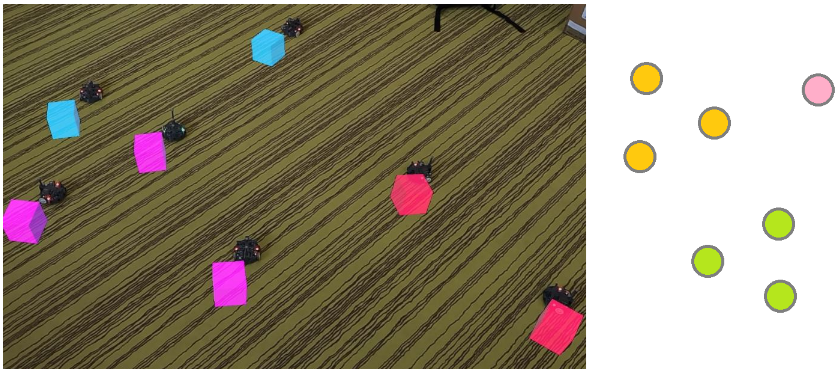
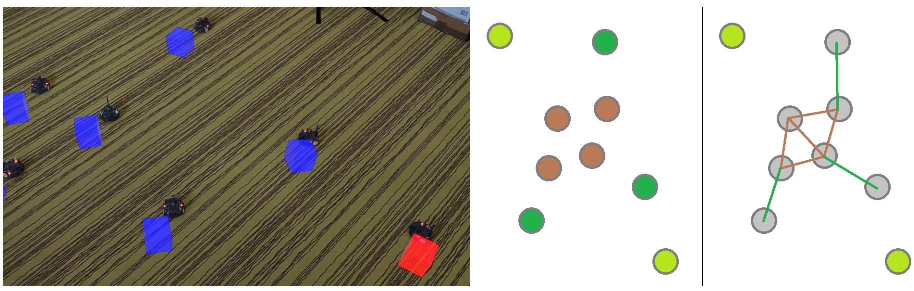


Fig. 16: Community

Allows a user to visualize at a glance the different robot groups with color-coded cubes, which can be useful to see if a fracture has occurred, but only allows the user to see it after it has occurred [8, 12, 14] (see Fig. 16). The cubes are opaque and may interfere with other visualizations, so playing with transparency or simply surrounding robots in the same group with color might improve this visualization.

- Leaves, Branches, and Trunk



(a) (b)

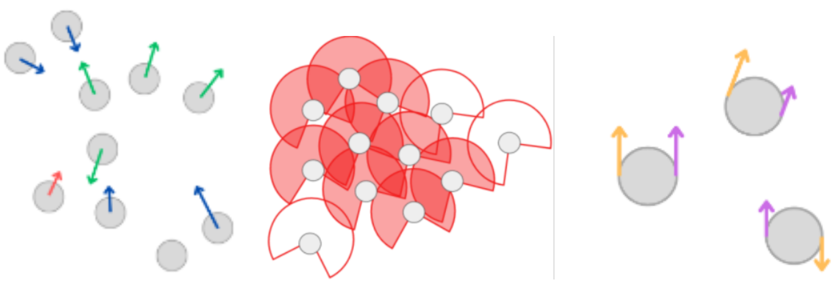
Fig. 17: (a) Leaves, Branches, and Trunk (b) Visualization idea

Allows a user to separate the agents into 3 categories. The Leaves are agents that do not share any link with any other agents, the Branches are agents from branches of the graph, there is only one way to reach them and the graph ends at the end of it. And last but not least, the Trunk corresponds to the last agents (see Fig. 17(a)). This visualization can be confused with the community one and the user may not understand which agents are the leaves, branches, or trunk. Since this visualization uses a graph, an improvement would be to represent the leaves with a cube as before but to connect the trunk agents with a line of one color and the branch agents with another (see Fig. 17(b)).

3.3 Discussion

To sum up, I mainly worked on setting up the augmented reality environment in the ARTUISIS platform, i.e. adding a data exchange between the augmented reality headset and the gateway software, and calibrating the virtual environment so that visualizations are located over the robots. The augmented reality application can be used in simulation mode or in a motion capture environment. As for the visualizations implemented, they were chosen because they are simple and only allow visualizing simple swarm data and are a first step towards more complex visualizations.

In order to understand and capture swarm behavior, future work will focus on determining the most useful visualizations for an operator. More useful visualizations could show the robots' field of perception [6, 15, 16], the force that dominates a robot's behavior, or the speeds of each wheel using arrows [6] (see Fig. 18). These visualizations should help an operator to apprehend fractures.



(a) (b) (c)

Fig. 18: Representation of (a) Dominant force (b) Perception and (c) Wheel speeds

Another future work will be to evaluate the relevance of these visualizations using user experiments but carrying out an experiment with a swarm of robots is no easy task as the experimental conditions must be the same for each participant, that is to say, the swarm must behave in the same way at each evaluation. To do this, it is better to use the simulation environment and then create several situation scripts. Those situation scripts could show cases of swarm splits or not with different swarming and flocking algorithms. The participant should indicate whether the swarm has broken up or not, once with the help of visualizations and once without.

4 Summary

4.1 Challenges

I encountered several difficulties during this internship. The first concerns my work organization as I have difficulty concentrating and focusing on a task and being generally organized. This caused me a few problems, including the feeling that the project was not progressing. Nonetheless, the weekly updates as well as my supervisors helped me overcome that feeling.

On the development side, I had some difficulties while programming the network and the code for the calibration.

For the network, I had some difficulties while testing the data exchange between the AR headset and the gateway. The AR headset sent and received its messages and I did not understand why. In simulation mode, the robots connect using a simulated IP address and I used the same one to connect the AR headset, with a different connection message. It did not work so I used a third simulated IP address, only for AR headsets, the problem still continued and I decided to use two sockets, one for sending messages and the other for receiving them. It worked and the code is cleaner too.

Another difficulty I encountered was managing the frequency with which messages were sent. The AR headset received messages with the marker data, but the robot data every two minutes. All the messages were sent from the gateway, but the AR headset only received the last message sent, so I decided to add a pause between each robot message sent. I also parallelized the code for sending messages between the gateway and the robots and the code for sending messages between the gateway and the AR headset.

For the calibration, figuring out how to get my two environments to have the same frame of reference was difficult for me. I set up several ways of calibrating the environment until I found the most accurate one. The first was to use 3 small markers and detect them at the same time so that I could move my virtual environment at the position of one of the markers and use the other two for the rotation. This was not precise at all as the markers were too small and detecting them at the same time did not provide their position precisely enough in the virtual environment.

So I decided to use a single, larger marker that I detect once and then move around to create several marks in the environment. I then measured the average position and rotation error between the marker detected by the AR headset and the marker detected by OptiTrack, and moved my virtual environment by this average. This method was tedious and there was always a gap between the robots and the visualizations. With some help, I managed to use only two measurements to get a correct calibration, as described in section 3.2.2.

4.2 Personal assessment

This was my second internship in a research laboratory, but this one was more comprehensive and gave me a better insight into the research field.

I wrote a bibliography on the subject of my internship before I started, which was my first major research project. Although stressful, I enjoyed researching articles and, above all, writing the report.

I am a little disappointed with everything I have not been able to do. I wanted to go further and think about setting up an experiment but I can say that the research work is time-consuming but very interesting.

Being in an internship in a research laboratory, I had the opportunity to see several aspects of the research field. I took part in experiments by some doctoral students, I was able to attend various meetings and observe the work of the different research professions and I was lucky enough to attend the seminar of the Interaction division.

I am now sure that I want to work in the research field. My goal would be to complete a thesis, in the field of augmented reality or multi-agent systems, in order to become an associate professor. I would also like to try my hand at being a research engineer.

5 Conclusion

This internship was proposed by the Lab-STICC and more specifically by the INUIT team of the Interaction department. It was part of a thesis called ARTUISIS, and the aim of this project was to implement an immersive and localized data visualization of a swarm of robots in Augmented Reality.

In this report, I began by showing the related works on the representation of localized data in augmented reality and data visualization in augmented reality applied to a swarm of robots. The main visualization techniques to use for localized data visualizations are situated and embedded visualizations, which allow displaying of data that makes sense in its environment. The displayed data mainly concerns the robots' state and the visualizations used are simple, such as graphs or labels. I then showed my work on the network, calibration, and visualizations. The visualizations implemented are fairly straightforward and now need to be sorted out and improved to best meet operators' needs. Future work could take into account any constraints associated with augmented reality, such as occlusion or depth cues, which would allow for more immersive visualizations and fewer errors associated with depth perception.

References

[1] Denis Kalkofen et al. "Visualization Techniques for Augmented Reality". In: July 2011, pp. 65–98. DOI: 10.1007/978-1-4614-0064-6_3.

[2] Wesley Willett, Yvonne Jansen, and Pierre Dragicevic. "Embedded Data Representations". en. In: *IEEE Transactions on Visualization and Computer Graphics* 23.1 (Jan. 2017), pp. 461–470. ISSN: 1077-2626. DOI: 10.1109/TVCG.2016.2598608. URL: http://ieeexplore.ieee.org/document/7539328/ (visited on 02/06/2023).

[3] Fabrizio Ghiringhelli et al. "Interactive Augmented Reality for understanding and analyzing multi-robot systems". In: *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*. ISSN: 2153-0866. Sept. 2014, pp. 1195–1201. DOI: 10.1109/IROS.2014.6942709.

[4] Alan G. Millard et al. "ARDebug: An Augmented Reality Tool for Analysing and Debugging Swarm Robotic Systems". In: *Frontiers in Robotics and AI* 5 (2018). ISSN: 2296-9144. URL: https://www.frontiersin.org/articles/10.3389/frobt.2018.00087 (visited on 03/03/2023).

[5] Alexandra Wheeler et al. "Diagnosing Robotic Swarms (Dr. Swarm)". In: (Apr. 2019). URL: https://core.ac.uk/display/213005558?utm_source=pdf&utm_medium=banner&utm_campaign=pdf-decoration-v1 (visited on 03/03/2023).

[6] Kent J. Libby, Lingrui Zhong, and Noah M. Van Stralen. "Diagnosing Robotic Swarms 2 (Dr. Swarm2)". In: (2020). URL: https://core.ac.uk/display/343944144?utm_source=pdf&utm_medium=banner&utm_campaign=pdf-decoration-v1 (visited on 03/03/2023).

[7] Phillip Walker, Michael Lewis, and Katia Sycara. "The effect of display type on operator prediction of future swarm states". In: *2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. Oct. 2016, pp. 002521–002526. DOI: 10.1109/SMC.2016.7844619.

[8] Gaetan Podevijn, Rehan O'Grady, and Marco Dorigo. "Self-organised Feedback in Human Swarm Interaction". en. In: ().

[9] Takefumi Hiraki et al. "Phygital Field: An Integrated Field with Physical Robots and Digital Images Using Projection-Based Localization and Control Method". In: *SICE Journal of Control, Measurement, and System Integration* 11.4 (July 2018). Publisher: Taylor & Francis _eprint: https://doi.org/10.9746/jcmsi.11.302, pp. 302–311. ISSN: 1882-4889. DOI: 10.9746/jcmsi.11.302. URL: https://doi.org/10.9746/jcmsi.11.302 (visited on 08/16/2023).

[10] Eva Pajorová, Ladislav Hluchý, and Marek Masár. "3d visualization of the simulation research results of new method for cooperative of a group of mobile agents for swarm behavior simulation". In: *Proceedings of the 2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*. ASONAM '13. New York, NY, USA: Association for Computing Machinery, Aug. 2013, pp. 1275–1279. ISBN: 978-1-4503-2240-9. DOI: 10.1145/2492517.2500291. URL: https://dl.acm.org/doi/10.1145/2492517.2500291 (visited on 08/16/2023).

[11] Karina A. Roundtree, Matthew D. Manning, and Julie A. Adams. "Analysis of Human-Swarm Visualizations". en. In: *Proceedings of the Human Factors and Ergonomics Society Annual Meeting* 62.1 (Sept. 2018). Publisher: SAGE Publications Inc, pp. 287–291. ISSN: 2169-5067. DOI: 10.1177/1541931218621066. URL: https://doi.org/10.1177/1541931218621066 (visited on 08/16/2023).

[12] Sumeet Batra, John Klingner, and Nikolaus Correll. "Augmented reality for human–swarm interaction in a swarm-robotic chemistry simulation". en. In: *Artificial Life and Robotics* 27.2 (May 2022), pp. 407–415. ISSN: 1614-7456. DOI: 10.1007/s10015-022-00763-w. URL: https://doi.org/10.1007/s10015-022-00763-w (visited on 08/16/2023).

[13] Sarjana Oradiambalam Sachidanandam, Sara Honarvar, and Yancy Diaz-Mercado. "Effectiveness of Augmented Reality for Human Swarm Interactions". In: *2022 International Conference on Robotics and Automation (ICRA)*. May 2022, pp. 11258–11264. DOI: 10.1109/ICRA46639.2022.9812080.

[14] Mingxuan Chen et al. "A multichannel human-swarm robot interaction system in augmented reality". In: *Virtual Reality & Intelligent Hardware* 2.6 (Dec. 2020), pp. 518–533. ISSN: 2096-5796. DOI: 10.1016/j.vrih.2020.05.006. URL: https://www.sciencedirect.com/science/article/pii/S2096579620300905 (visited on 08/16/2023).

[15] Masanori Sugimoto et al. "RoboTable2: a novel programming environment using physical robots on a tabletop platform". In: *Proceedings of the 8th International Conference on Advances in Computer Entertainment Technology*. ACE '11. New York, NY, USA: Association for Computing Machinery, Nov. 2011, pp. 1–8. ISBN: 978-1-4503-0827-4. DOI: 10.1145/2071423.2071436. URL: https://dl.acm.org/doi/10.1145/2071423.2071436 (visited on 08/16/2023).

[16] Ana Maria Rubio Denniss, Thomas E. Gorochowski, and Sabine Hauert. "Augmented reality for the engineering of collective behaviours in microsystems". In: *2019 International Conference on Manipulation, Automation and Robotics at Small Scales (MARSS)*. July 2019, pp. 1–6. DOI: 10.1109/MARSS.2019.8860907.