

WROCŁAW UNIVERSITY OF SCIENCE AND TECHNOLOGY
FACULTY OF ELECTRONICS

FIELD: Control Engineering and Robotics (AIR)
SPECIALIZATION: Embedded Robotics (AER)

MASTER OF SCIENCE THESIS

Augmented reality goggles in robotic applications

Zastosowanie okularów rozszerzonej rzeczywistości
w aplikacjach robotycznych

AUTHOR:
Dawid Śliwa

SUPERVISOR:
dr inż. Janusz Jakubiak, W4K7

GRADE:

Contents

1	Introduction	3
1.1	Purpose and scope of work	3
2	Introduction to Augmented Reality	5
2.1	Technology overview	5
2.1.1	Types of image projection	6
2.1.2	Positioning and location	7
2.1.3	User control and interaction	10
2.2	Products available on market	11
2.2.1	Software solutions	11
2.2.2	Hardware devices	11
3	Applications of Augmented Reality	13
3.1	ARDebug	13
3.2	Virtual sensing technology	13
3.3	Interactive Spatial AR in Collaborative Robot Programming	15
4	Implementation of AR in robotics	17
4.1	Used technologies	17
4.1.1	3D Engine	17
4.1.2	Vuforia	18
4.1.3	Robotic Operating System	19
4.2	Test results	20
4.2.1	Robotic arm	20
4.2.2	Mobile robot	22
5	Summary	25
	Bibliography	25

Chapter 1

Introduction

Robots become more and more often seen in our environment. Starting from nowadays standard industrial applications and ending with home appliances robots. They all have more or less user friendly interfaces created to program or control them. In factories stationary or hand-held controllers are most commonly used, whereas in consumer applications the smartphone is most often used. The problem is that this kind of interaction is not natural for people. For comparison, communication between two employees working together is mostly done by voice, gestures and sometimes touch. That is why modern controllers should be using these. This could improve a way of interaction on human-machine level.

A few years ago, a revolution called Industry 4.0 began which most important statement was to not replace peoples in factories by machines, but allow them to cooperate at production line. From that time companies try to simplify teaching process of robots and give them ability to sense the changing environment. Also enhancements are done on the other side. Employees are equipped with many solutions with extend their perception. They allow to better understand of what machines do or even see what they "think".

1.1 Purpose and scope of work

This thesis focus on Augmented Reality and its usage in modern factories and research facilities. At the beginning, different types of AR technologies are compared to give an overall view on how this devices works. Then industrial and commercial products available currently on the market are presented. The last part of topic studies present selected solutions which are already used in real world applications.

Research part of this thesis presents simple examples of implementation of the AR in robotic applications. The topic works covers the issue of planning movement of robotic arm and also controlling and presenting data from a mobile robot. This should give more or less understanding what this technology is capable of and what are its current limitations.

Chapter 2

Introduction to Augmented Reality

The perception of our surroundings is made to a large extent by the organ of sight. Thanks to that we are able to navigate and operate in our real environment. But what if we try to trick it by placing displays in front of our eyes? Depending on content generated by computer it could simply show some additional information or create illusion of being completely somewhere else. To distinguish types of immersion the concept of a "Reality-Virtuality Continuum" [26] was created. Its graphical representation is shown in Figure 2.1.

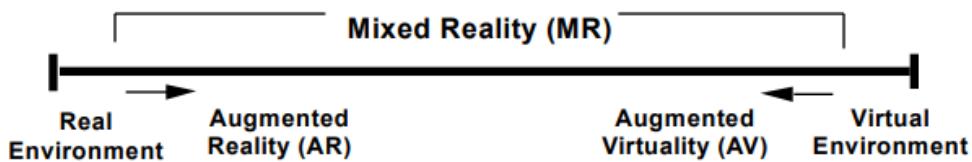


Figure 2.1: Reality-Virtuality (RV) Continuum [26]

On the most left side of this line there is our real environment with real objects in no way disturbed by computer graphics. On the opposite side there is Virtual Reality with fully 3D generated world. Between two of those is everything which is mixing one part with another. Depending on what is the balance between them, we talk about Augmented Reality or Augmented Virtuality. For example when an operator is receiving some simulated cues to augment his natural feedback then it is called AR. When in virtual world appears some real life objects or persons then it is called AV. Good example for that are modern news where reporters work at green screen and in television they appear in 3D generated studios. As this paper discusses topic of using Augmented Reality in robotics applications that is why technical aspects of Augmented Virtuality and Virtual Reality will not be considered.

2.1 Technology overview

AR devices can take many different forms but a way of processing data is almost always the same. In the simplest case could be explained as the following process. At first a device needs to capture an image and localize itself or the user in environment then mix those data with CG objects and at the end to display them on a screen. Figure 2.2 presents this pipeline in graphical form with few additional steps.

This section shows the different types of technologies that correspond to the selected elements of this process.



Figure 2.2: Simplified AR pipeline [24]

2.1.1 Types of image projection

There are several ways of displaying virtual objects in our environment. Depending on its type they could be more or less immersive for the user.

The most popular and the simplest type of projections is video-mixing. It involves the usage of devices equipped with a camera and standard display to present AR content. In the most common cases, smartphones or tablets are used because they have all the necessary components built in and are very portable. Figure 2.3 presents an example of such projection. The great advantage of this approach compared to others is that you can use one device in cooperation with other people, which could result in a significant reduction of operating costs. The disadvantages should be mentioned that the employee is not able to observe the environment and use both hands to do his job. Therefore, this type of projection is most often used in devices used to supervise the operation of machines.

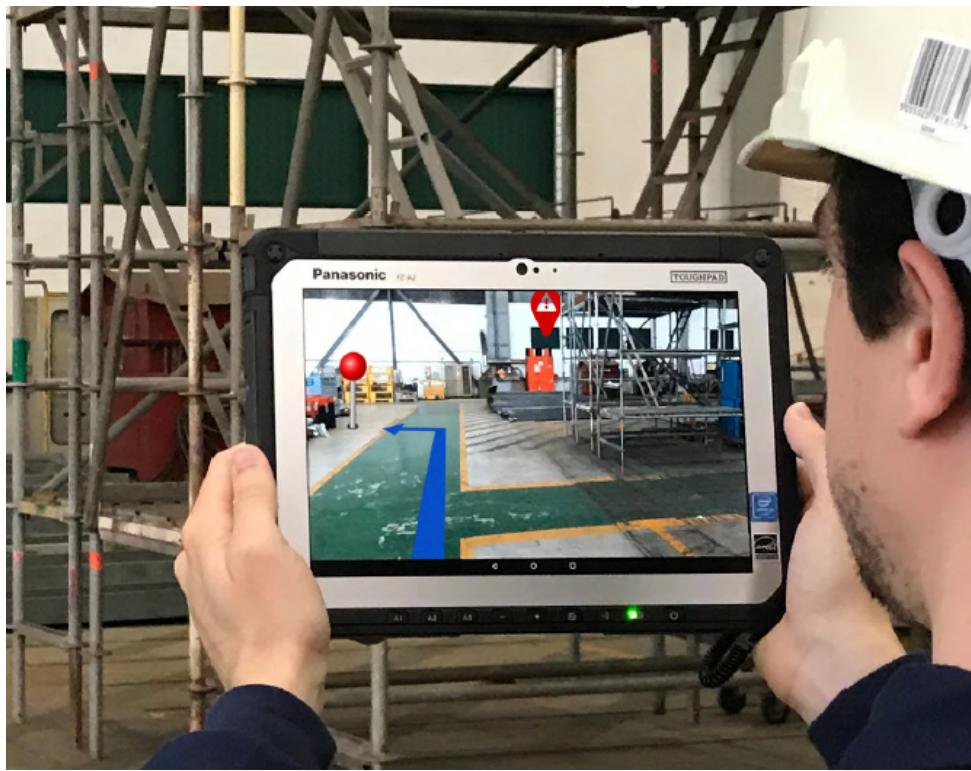
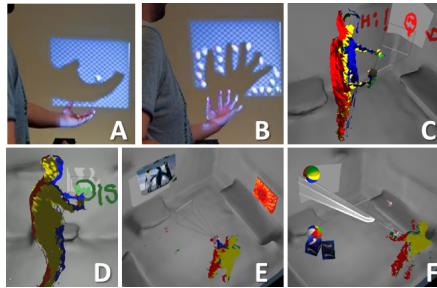


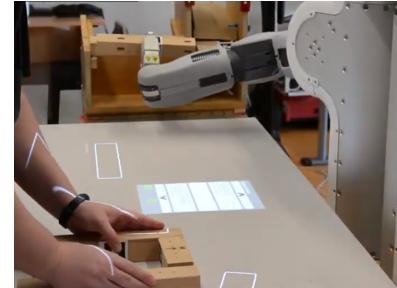
Figure 2.3: Worker using tablet as video-mixing device [24]

Second type of projection is slightly different than first one. It is called spatial display. In this case virtual objects are shown directly on real environment surfaces by usage of digital projectors. It could be realized in two ways. We could have hand-held device with a projector that will work similar to flashlight or a stationary mounted projector. In the first case we could for example inspect virtual paths of transportation robots in warehouses or factory by highlighting floors. Also autonomous cars can use their headlights to display information on the road and for instance to inform pedestrian that they could safely cross the passage.

Stationary projectors are most often used with collaborative robots. They can display for example some cues for workers which item they should pick-up or where place in the shared workspace. Presented types of projection can be seen in Figure 2.4.



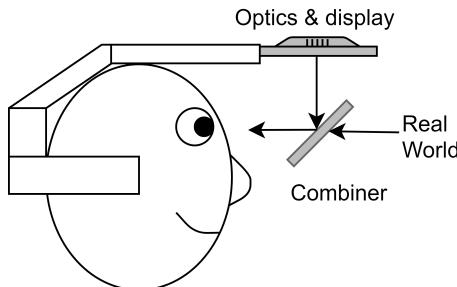
(a) Augmented projector [28]



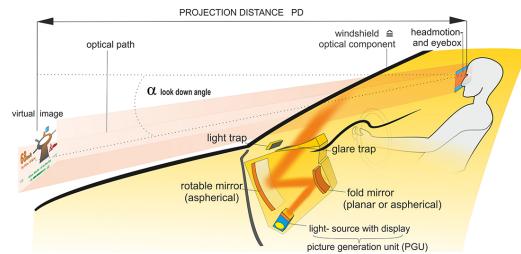
(b) Man and robot coworking [25]

Figure 2.4: Examples of spatial projection

Next AR display technology is optical see-through. Its principle of operation boils down to projecting images on partially reflective surface. This allow to combine real world view with generated graphics which appear as holograms floating in space. There are two ways to achieve this goal. Device could use a head-mounted or head-up displays. Figure 2.5 shows how the projection is carried out with particular case. First type presented at image 2.5a is commonly used in situations where worker need to have both hands free to do his work. At HMD some step by step instructions can be shown or in case operating with robots, live sensors data. The example from image 2.5b shows the most common scenario of usage of HUD. Nowadays they are showing information about speed and navigation guidance for a driver but in autonomous vehicles they could display also some cues what the car sees and what it intends to do.



(a) Head-Mounted Display



(b) Head-Up Display [5]

Figure 2.5: Examples of optical see-through projections

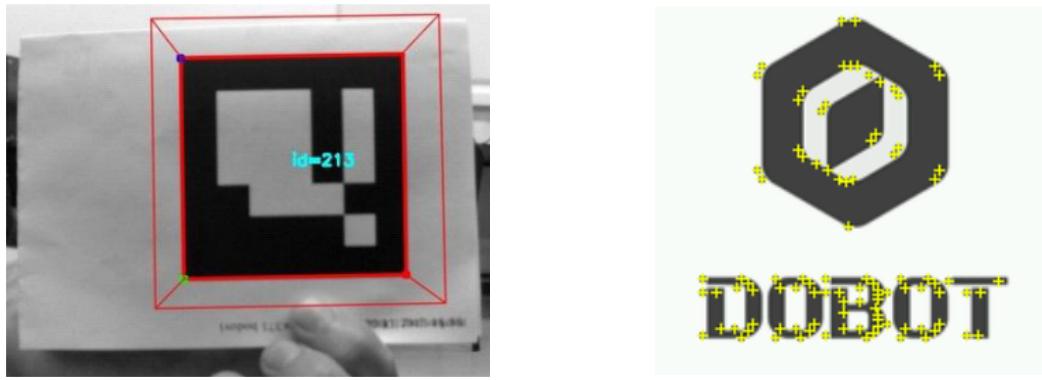
2.1.2 Positioning and location

Only small part of AR devices is presenting data that does not depend on real environment obstacles. These include basic versions of HUD and smartglasses. The rest of them needs to somehow localize itself or detects position of the user. In this section various types of positioning techniques will be presented.

Marker-based localization is the simplest one. The principle of operation consists in detection of predefined shape or image by the device camera. To achieve this first of all

data need to be preprocessed. The most trivial example is when our marker have square shape and high contrast. In that case the algorithm needs only to know what is length of the side. With this information it is able to calculate position and orientation of the device basing on detected corners. As the most important are only the edges the center can have any shape. It allows to store some information inside and also help to determinate from which side of square the camera is looking. Figure 2.6a shows the detected marker with the read ID number assigned to it. Pros of this approach is quite low requirement for computational power and a easy way to generate many unique ID's. The disadvantage is that the user must print these images and arrange them in chosen locations.

There is also a possibility to track normal images or even 3D objects. In this case there are algorithms used extracting some particular features from provided data (Figure 2.6b). Basing on them device could be able to localize itself in space only by looking on for example machine logo. Pros of this approach is better tracking of object even when whole image is not in camera view. Disadvantage is more complex computation.



(a) Detected ArUco marker with ID [22]

(b) Image with marked extracted features

Figure 2.6: Example of markers used for localization purposes

Next type of localization is based on measurements obtained via external devices. There are many possible implementations of this approach but they can be categorized based on the principle of operation.

Optical tracking of passive markers [23] is best known in film industry. It is used in Motion Capture studios to transfer movement of actor to 3D modeled characters. It requires usage of multiple high-speed cameras with infrared illuminators fixed around the measurement area to triangulate a reflective marker position. Successful capture of tracked point by at least two camera give a sub-millimeter precision. To avoid situation where marker is occluded by some obstacles the localized object could be equipped with redundant markers. A disadvantage of such system is limitation of operating area caused by strength of the reflected light. There is also a way to increase to a certain extent system range by using active markers which are light sources but this causes problems with a need to provide power to those.

From optical localization method there are also systems working without any markers. They are based on 3D depth cameras which provide not only image but also give information about distance from objects. This allows not only get position of the tracked device but also could give feedback about scanned environment. This kind of systems generally have operation distance from 0.5 to 8 meters. Its precision of positioning is inversely proportional to it.

External localization could be also realized by measuring field strength or time of flight of electromagnetic waves. In this case at least three transmitters are needed to establish

position. This method is slightly less accurate in comparison to optical ones but have a huge advantage in the form of being able to track devices without direct visibility. Also the range and refresh rate of such systems is significantly larger than of optical methods.

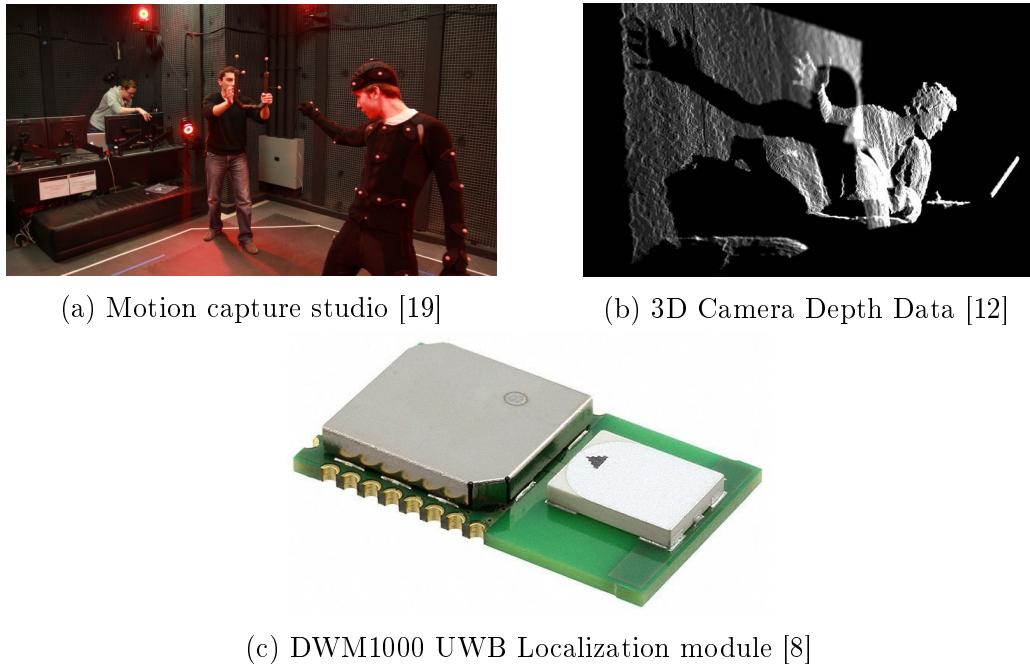


Figure 2.7: External localization systems

Last presented type of positioning systems is Simultaneous Localization And Mapping (SLAM). This technique uses Laser Range Sensors or 3D Depth cameras to create map of environment. Basing on that data algorithm could calculate position and orientation in space. The big advantage of this solution is the ability to integrate all elements into one device which makes it almost limitless in terms of working area. Also additional advantage is that the obtained data can be used to classify objects located in the environment. Unfortunately, such calculations take huge amount of processing power so a device require more expensive components to be able run in real-time. In Figure 2.8 can be seen 3D map of room created by using depth sensor and SLAM algorithm.

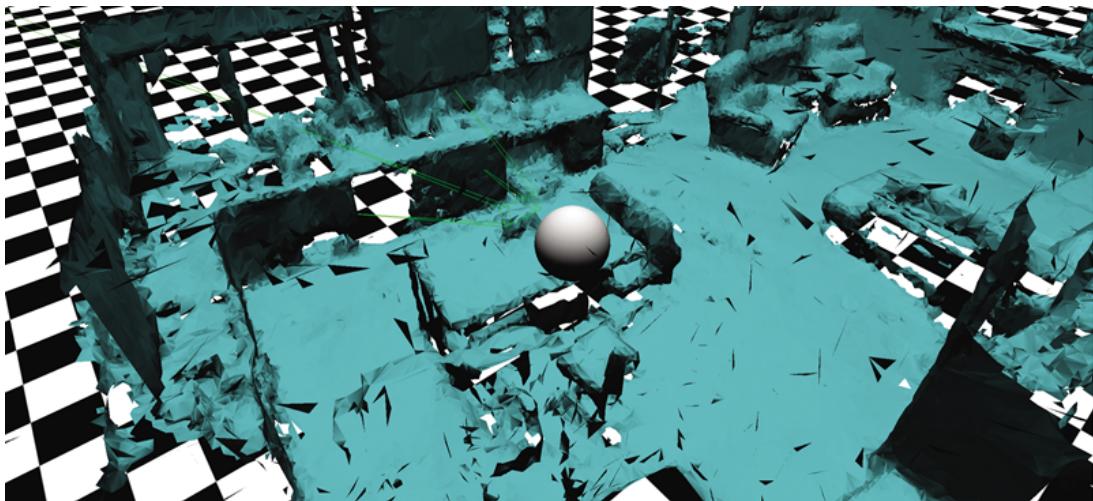


Figure 2.8: Room mapped using SLAM algorithm with usage of Microsoft Hololens

2.1.3 User control and interaction

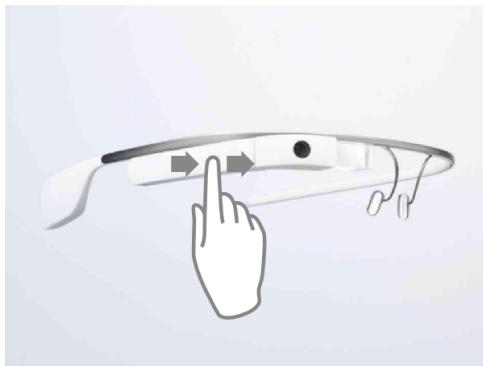
Augmented reality devices require a different approach to how the user interacts with them. As the worker is no longer limited by his stationary desk there is a need to change the way of interaction. This section discusses most commonly used methods.

The simplest HMD devices, where in the field of view we have only displayed text or images are using the simplest interaction solutions. In most cases there are buttons or small touchpad placed in the frame (Figure 2.9a) which allow user to control screen content. This approach allows to reduce production costs so that it can have lower price tag and get to more customers.

When device provides ability to observe 3D objects in space then the user needs some controllers to be able to operate it. On the market the most popular ones use optical tracking (Figure 2.9b) but there are also devices which use electromagnetic field. In addition, all are equipped with 6-DOF sensors to enhance stability of calculated position. This type of input is very precise but requires holding the controller which in some cases may hinder normal work.

The solution to this problem is to recognize gestures and track them (Figure 2.9c). This allows to operate with 3D environment while still having both hands free. Detection is carried out using depth cameras. Unfortunately like in case of using SLAM algorithms this method consumes a lot of computing power so it is reserved only for the most expensive devices.

The last type of interaction is voice control. It is a very intricate topic due to diversity of dialects and accents. With current technology embedded computers can interpret only single words or predefined sentences however, using the resources of cloud services it is possible to extract information from context using natural language processing (NLP) algorithms. The disadvantage of this solution is a requirement of continuous access to the Internet and quite slow response.



(a) Touchpad at Google Glass [11]



(b) Mixed Reality Controllers [16]



(c) Using gesture with Microsoft Hololens [15]

2.2 Products available on market

Augmented reality is still quite young technology. Omitting the fact that it was used in the army and civil aviation since the 1960s, it is now becoming available to industry and individual users. There are more and more companies trying their strength in this sector of the market. They use a variety of image displays and interaction technologies, and develop newer and newer solutions. This section will show few chosen AR devices and software.

2.2.1 Software solutions

The cheapest entry level to AR world is use of dedicated SDK's on computer, smartphone or tablet. On the market there are many ready to use solutions which cover different use cases. Most of them are commercial products, but there are also some open-source one. Table 2.1 shows the collected information about several of them.

2.2.2 Hardware devices

The devices presented at table 2.2 are currently focused mainly on B2B cooperation. High prices and low maturity of the technology to the consumer market have an impact on this. Also there is no one unified environment between them so all applications are written to fulfill specific case. However, some devices can be bought in normal sales, so it is not a completely closed market for a regular customer.

SDK Name	Licence	Platform	Capabilities	Description
ALVAR [1]	GPL	Mobile: Android, iOS, UWP Desktop: Linux, Windows, Mac	Tracks 2D targets Multi-marker tracking Markerless, feature-based 3D tracking Extended tracking, SLAM, sensor fusion	General purpose augmented reality library basing on OpenCV. Developed by VTT Technical Research Center of Finland.
ARKit [3]	Commercial	Mobile: iOS	2D Object Detection and Tracking Shared AR Experiences Detect known 3D objects Face Tracking	Library developed by Apple company. Available only for devices with Mac OS and iOS.
ARCore [2]	Apache 2.0	Mobile: Android, iOS Desktop: Linux, Windows, Mac	Motion tracking Markerless, feature-based 3D tracking Environmental understanding Light estimation	Google's platform for building augmented reality experiences. Works on Android phones and tablets running Android 7.0 and later or PC.
ArUco [22]	GPLv3	Mobile: Android Desktop: Linux, Windows, Mac	Detect markers: ARUCO, AprilTag, ArToolKit+, ARTAG, CHILITAGS	ArUco is an OpenSource library for camera pose estimation using squared markers. ArUco is written in C++ and is extremely fast.
Vuforia [21]	Commercial	Mobile: Android, iOS, UWP Desktop: Linux, Windows, Mac	Model 2D/3D Targets with Deep Learning Recognize multiple objects, from multiple angles VISLAM for markerless AR External Camera for iOS	Vuforia offers dynamic Object Recognition and can perceive images, 3D models and environments to provide development flexibility. It has also great community and many tutorials.

Table 2.1: Most popular SDK's on augmented reality market

Device name	Type of projection	Hardware	Sensors	Connectivity	Battery (life time)	Mechanical parameters	User input	Operating system	Capabilities	Price
Microsoft Hololens [17]	Optical see-through 1320x1200 px per eye 40 degrees view angle	Intel x86 CPU Holographic Processing Unit 2 GB RAM 64 GB storage	Depth Sensor Camera 2MP Video Camera 4x environment understanding cameras IMU Ambient Light Sensor Microphone On-board Speakers	Wi-Fi 802.11ac Bluetooth® 4.1 LE 3.5mm Jack micro-USB	4500 mAh 2-3 hours	Weight: 579 g Adjustable head band	Gesture input Wireless clicker Voice support Buttons on frame	Windows™ 10	Spatial sound Gaze tracking 6-DOF tracking Mapping of environment Teleoperation	\$4500
Epson MOVERIO PRO BT-2000/2200 [9]	Optical see-through 960x470 px per eye 23 degrees view angle	TI OMAP 4460 1.2 GHz 1 GB RAM 8 GB internal storage 32 GB external microSD	5MP Stereo Camera GPS IMU Ambient Light Sensor Geomagnetic Sensor Microphone	Wi-Fi 802.11 b/g/n/a Bluetooth® 3.0 & BLE 3.5mm Jack micro-USB	2x 1240 mAh ~4 hours	Weight: 270-290 g Depending on version: - adjustable head band - rubber band	Wired controller Voice support	Android™ 4.0.4	Teleoperation 6-DOF tracking Task guidance	\$2000
DAQRI Smart Glasses [6]	Optical see-through 1300x768 px per eye 44 degrees view angle	6th Intel® Core™ m7 Dedicated Vision processing unit 64 GB storage	Color Camera AR Tracking Camera Depth Sensor Camera IMU 2 Microphones with Active Noise Cancellation	Wi-Fi 802.11 a/b/g/n/ac Bluetooth® 4.1 LE 2 USB 3.1 Type C Ports 3.5mm Jack	5800 mAh	Smart Glasses: 355 g Compute Pack: 496 g Adjustable head band	Gesture input Voice support	Custom solution	Mapping of environment Teleoperation 6-DOF tracking	Unknown
ARVIND SANEEV LUMEN [4]	Projection based 640x360px screen size: 10"-11.2"	Broadcom BCM2835 GHz 512 MB RAM 32 GB external microSD	Color Camera Depth Sensor Camera	Wi-Fi 802.11 b/g/n Bluetooth® 4.1 LE micro-USB	1800 mAh ~30 min	Unknown	Gesture input	Linux	Objects recognition Mapping of environment	Unknown
Light Guide Systems Classic [13]	Projection based	Any PC	Color camera	HDMI USB	AC Powered	Unknown	Camera feedback	Windows™	Augmentation of workstation Task guidance Automatic part assembly check Production time recording	Unknown
Magic Leap One [14]	Optical see-through 1280x960 px per eye 50 degrees view angle	CPU: NVIDIA® Parker GPU: NVIDIA Pascal™ 8 GB RAM 128 GB Storage	Depth camera Environment understanding cameras IMU Microphone On-board Speakers	Wi-Fi 802.11ac/b/g/n Bluetooth® 4.2 USB-C	~3 hours	Weight: 345 g	Gesture input 6-DOF tracked controllers Voice support	Lumin OS	Mapping of environment Teleoperation 6-DOF tracking	\$2,295
Google Glass Enterprise Edition [10]	Optical see-through 640x30 px	Intel Atom processor 2 GB RAM 32 GB storage	5 MP camera IMU Ambient Light Sensor GPS & GLONASS Barometer Bone conduction transducer	Wi-Fi 802.11n/ac Bluetooth®	780 mAh ~8 hours	Weight: 36g	Voice commands Touched Mobile app	Android™ 4.4	Teleoperation Assistant Task guidance	\$1,750

Table 2.2: List of selected AR goggles models

Chapter 3

Applications of Augmented Reality

This chapter presents some found on the internet publications about using AR in robotics applications. It should allow to acquaint with the possibilities of this technology and suggest what areas have not been researched yet.

3.1 ARDebug

The first issue raised will be the use of AR to visualize the parameters of individual robotic swarm units basing on research paper [27] created by scientists from York Robotics Laboratory. The tool presented by them could help easier to find bugs in implemented algorithms by giving real-time visual feedback based on each robot state.

The configuration of the environment requires the use of a locating system. Researchers used for this purposes a 5 MP camera which was placed 2.5 m above the center of operating area and the ArUco markers placed on robots. Then, using the algorithms built into the OpenCV library, they determine their position. The data obtained this way was packed in JSON structure which is accepted by ARDebug application. Authors deliberately chose this format to allow easy integration other location methods.

Another element required by the application is data of the status of each robot. They also must be provided in JSON format over Wi-Fi or Bluetooth to the supervising computer.

After collecting all the information ARDebug imposes on video from camera, data acquired from tracking system and robot. Additionally, it presents them in the form of tables with key value representation and graphs. Example view from this program is presented in Figure 3.1.

3.2 Virtual sensing technology

Another issue was also created to improve and speed up the testing of algorithms written for the purpose of robotic swarms. This time, the research work [29] of scientists from Universit Libre de Bruxelles was examined. In this paper they describe created by them system ARGoS which allows to simulate sensors input without having installed real sensors. This approach significantly reduces the costs of testing a large number of robots by decreasing the amount of components needed per unit. In addition, it offers the possibility of flexible configuration of the sensors work range without any hardware modifications.

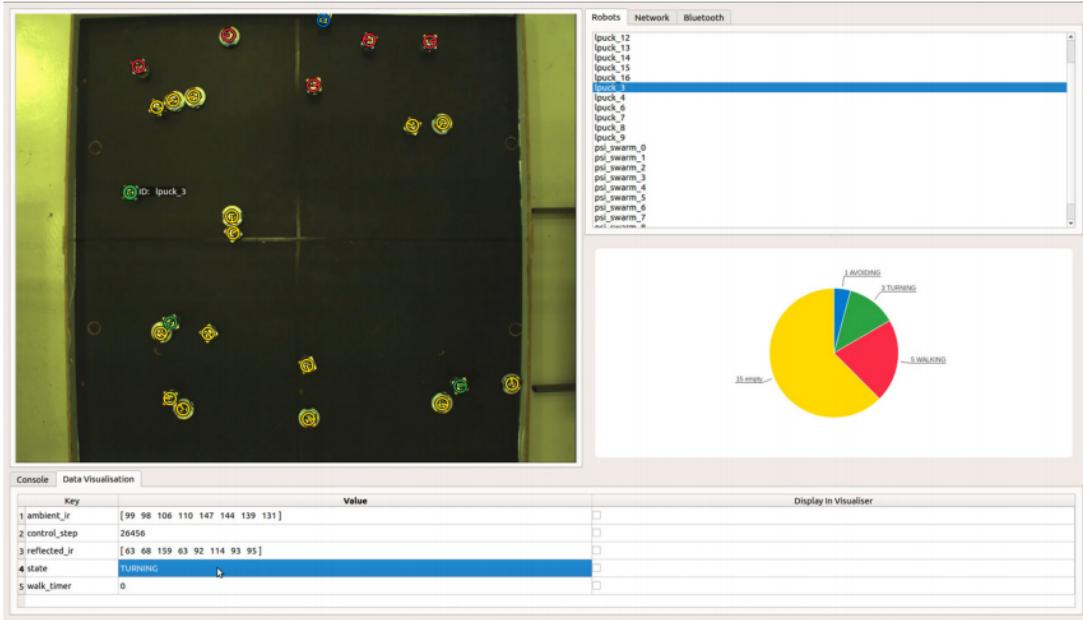


Figure 3.1: The ARDebug user interface [27]

Like in the previous system there is also need of reading robots positions. It was realized in the same way, by using a camera suspended above the working area and markers on tracked units. The data collected in this way is used to reproduce the scene in the simulator. Then virtual sensor modules calculate their outputs and provide wireless this information to real robots.

Whole process could be viewed in three perspectives. First one is virtual environment, second AR video-mixing and third by lighting LEDs on real robots. Output of the system can be seen in Figure 3.2.

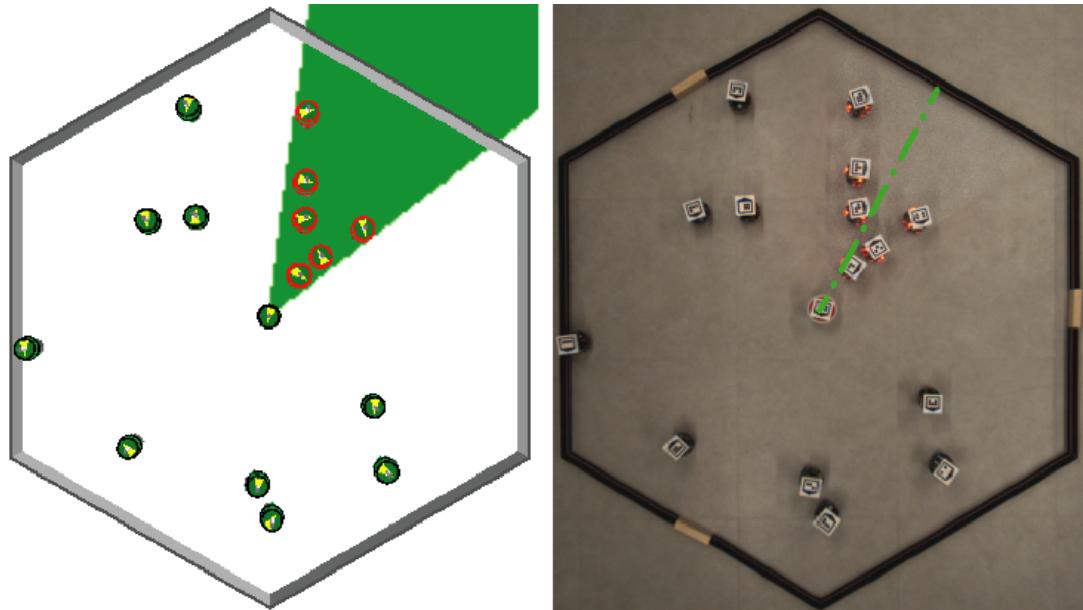


Figure 3.2: The ARGoS sensor simulator [29]

3.3 Interactive Spatial AR in Collaborative Robot Programming

Scientists from Brno University of Technology decided to explore the subject of human cooperation with machines. In their paper [25] they described a concept of utilization of shared workspace as an interactive space which could be used to visualize and control movements of collaborative robot. The goal was to create a system which will simplify programming to such level that every ordinary skilled worker could customize robot's behavior and adopt it to current work or preferences.

To test different solutions the researchers have created an easy to deploy and modular stand. It consisted of standard workshop table with capacitive touch foil on it, PR2 collaborative robot, Microsoft Kinect V2 depth sensors, two speakers and truss with projector hanging on it. Complete setup can be seen in Figure 3.3.



Figure 3.3: Setup of collaborative workshop desk [25]

Prepared elements change shared workspace into interactive display. Thanks to this, employee can mark the work areas of the robot in a very intuitive way and give him instructions by using simple GUI. Also this approach allows him to continuously observe what decisions robot is making without looking away from his job.

The authors carried out research on a group of people to check whether the applied solution introduces a sense of safety when working with collaborative robots. It was found

out that almost every participant assessed positively their work, however they also point out that GUI should be improved.

Chapter 4

Implementation of AR in robotics

The aim of the study was to investigate the possibility of using augmented reality in robotics. For this purpose, several issues related to stationary and mobile robots were selected.

In the first case, it was decided to create a user interface that would allow to remotely control the position of the robotic arm using gestures. Thanks to this, the operator is able to set a trajectory of motion along the points, working from a safe distance and simultaneously observing the behavior of the machine.

In the case of mobile robots, the topics of visualization of sensory data and route planning were examined. Displaying in real time sensors readings in user environment could help with debugging some hardware or software issues. Also being able to see waypoints of a robot on a floor can help to check if the robot is not going to hit any obstacle.

4.1 Used technologies

In order to implement selected tasks, it was decided to use the augmented reality goggles for this purpose. The choice fell on Microsoft Hololens due to the good performance, tracking quality and the ability to loan them for testing. Picture of them can be seen in Figure 4.1. Technical parameters was presented earlier in the table 2.2.



Figure 4.1: Picture presenting Microsoft Hololens AR Googles [18]

4.1.1 3D Engine

Application for Hololens could be written in every programming language which has support for Universal Windows Platform API. However, due to the fact that goggles are

working in a 3D environment, developer also need a graphics engine to be able to create spatial applications. On the market there are two main competitors which is Unreal Engine and Unity Engine. First one in a subjective sense, is more intuitive and provides a better visuals. In addition, it gives better control over code execution, due to possibility of writing backend in C++. Unfortunately it does not have good support from Microsoft side for Hololens that is why it was not chosen to do this tasks.

On the other hand, Unity thanks to a large community is easy to learn. Many tutorials helps programmers quite fast get familiar with user interface (Figure 4.2) and give ability to start deploying their first apps, even without writing a single line of code. However, for every non standard behavior a C# script file needs to be created.

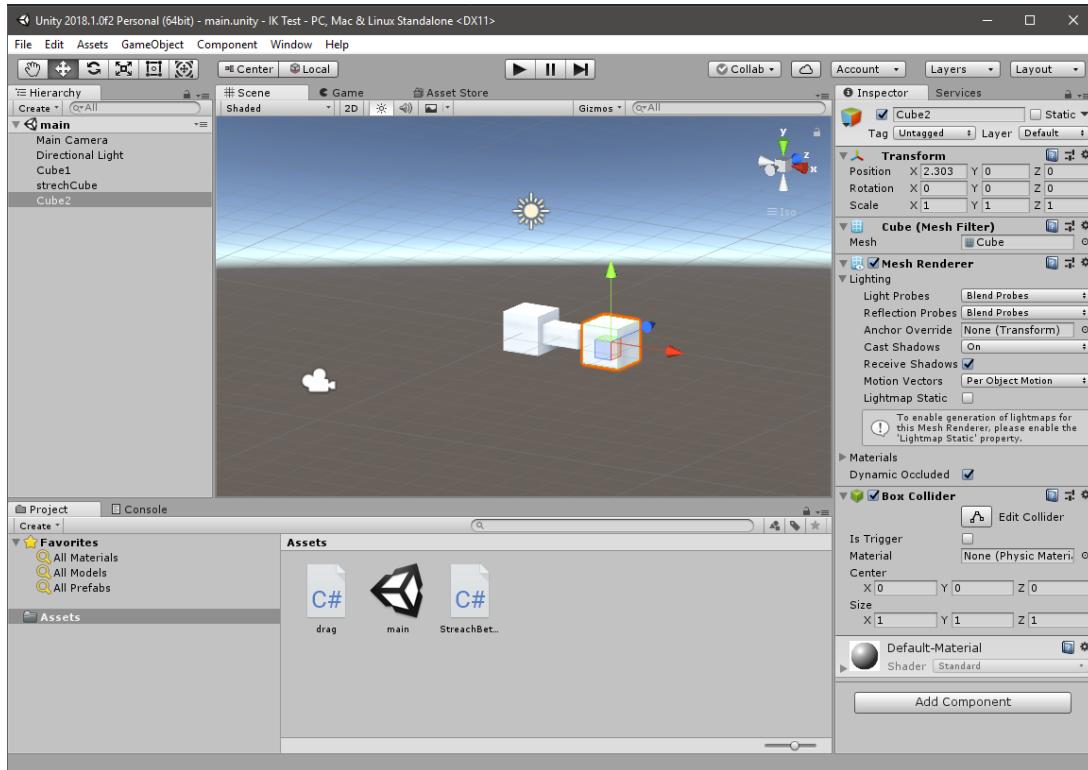


Figure 4.2: Unity Editor main window

Developing an applications for Microsoft Hololens in the Unity Engine has additional advantages in terms of the availability of the HoloToolkit library. It allows user to easily use advanced goggles functions such as location or gesture support by using already prepared assets or scripts. The only thing a programmer has to do is to pin the function he needs to the object.

4.1.2 Vuforia

Vuforia is an multiplatform Engine which uses advanced computer vision algorithms to bring AR experience to almost every kind of application. Using it, it is possible to locate and track position of device basing on the 2D markers/images or provided 3D models of real objects. Thanks to that user is able to see on the screen elements created in 3D environment in a real world. All features of the engine was presented in table 2.1

The decision to use this software was made for two reasons. The first one was easy integration with unity environment and second one was possibility to detect not only

predefined markers but any kind of image. In Figure 2.6b can be seen preprocessed Dobot robotic arm logo with extracted features, which is used in this research.

4.1.3 Robotic Operating System

Both of used in this research robots was controlled by Robotic Operating System. It is open-source platform which provides software libraries and tools for building robot applications.

The principle of ROS operation is based on the idea of running many nodes which each of it is representing some behavior. In example one of them could read some sensor data, process it and at the end share it with others. Communication system between nodes is based on asynchronous anonymous publish/subscribe mechanism (Figure 4.3). Thanks to that we get a flexible and modular platform which allows us to configure any data flow we want. However, not all operations can be performed in an asynchronous manner. To this end, ROS also provides a synchronous request/response mechanism called services.

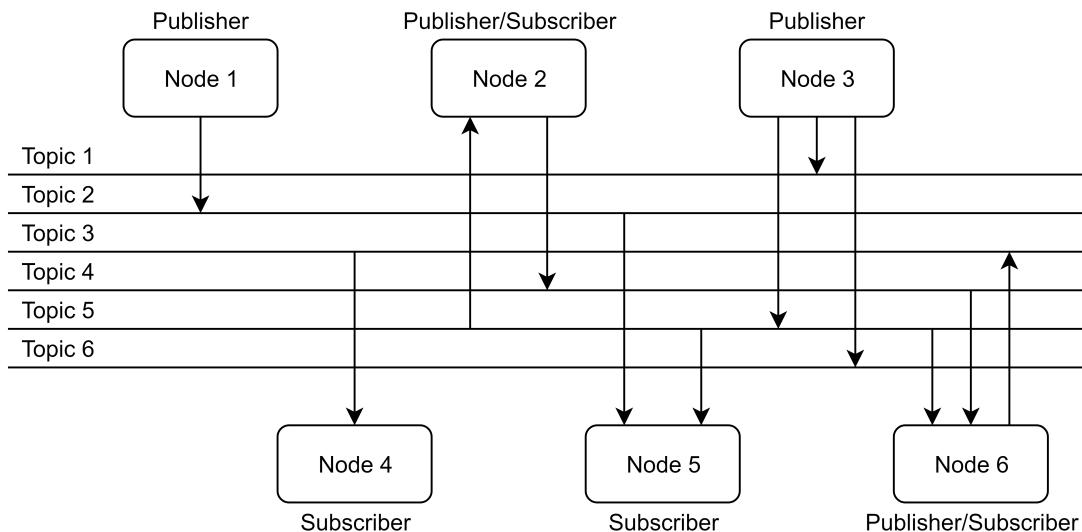


Figure 4.3: ROS publish/subscribe mechanism

Another great feature is also ability to record and playback messages. As the data in the system is shared by publish/subscribe mechanism there is easy way to capture them in example on real robots and then replay on offline machine without any changes in code. In combination with tools like rviz we are able to recreate and visualize all robots behaviors without having access to them.

Connection between ROS and external system could be realized by ROSBridge. This library is able to create server which could send and receive data in JSON format. Each frame is containing object name, topic/service name, messageType and optional payload. Such a simple solution allows for easy implementation of the client on any device. That is why this tool was used in the research to be able to exchange data between ROS and Unity Engine.

4.2 Test results

4.2.1 Robotic arm

The tools presented in the previous section were used to implement the task set at the beginning of the chapter. Thanks to the created application, the user is able to move the virtual object in space by means of gestures, which is then followed by the effector of the robotic arm. The selected positions could be memorized and in this way motion sequences could be created.

The robot used in this part is called Dobot Magician [7]. This a simple desk robotic arm with 3 degrees of freedom and the possibility of replacing the effectors. All axis are driven by stepper motors so before use in needs firstly perform homing action. The robot can be seen in Figure 4.4.



Figure 4.4: Dobot Magician robotic arm [7]

As mentioned before, the application was created in Unity using the Holotoolkit library. As far as the supplied elements allowed for trouble-free operation of the peripherals provided by goggle sensors, connecting them with ROS required adding one's own module. It was written as a C# script, in which the websocket connection to the ROSBridge server was opened. Then there was need to create protocol layer which will pack data

into proper JSON messages. Their use allowed creating two-way communication between the robot and the application in Unity (Figure 4.5).

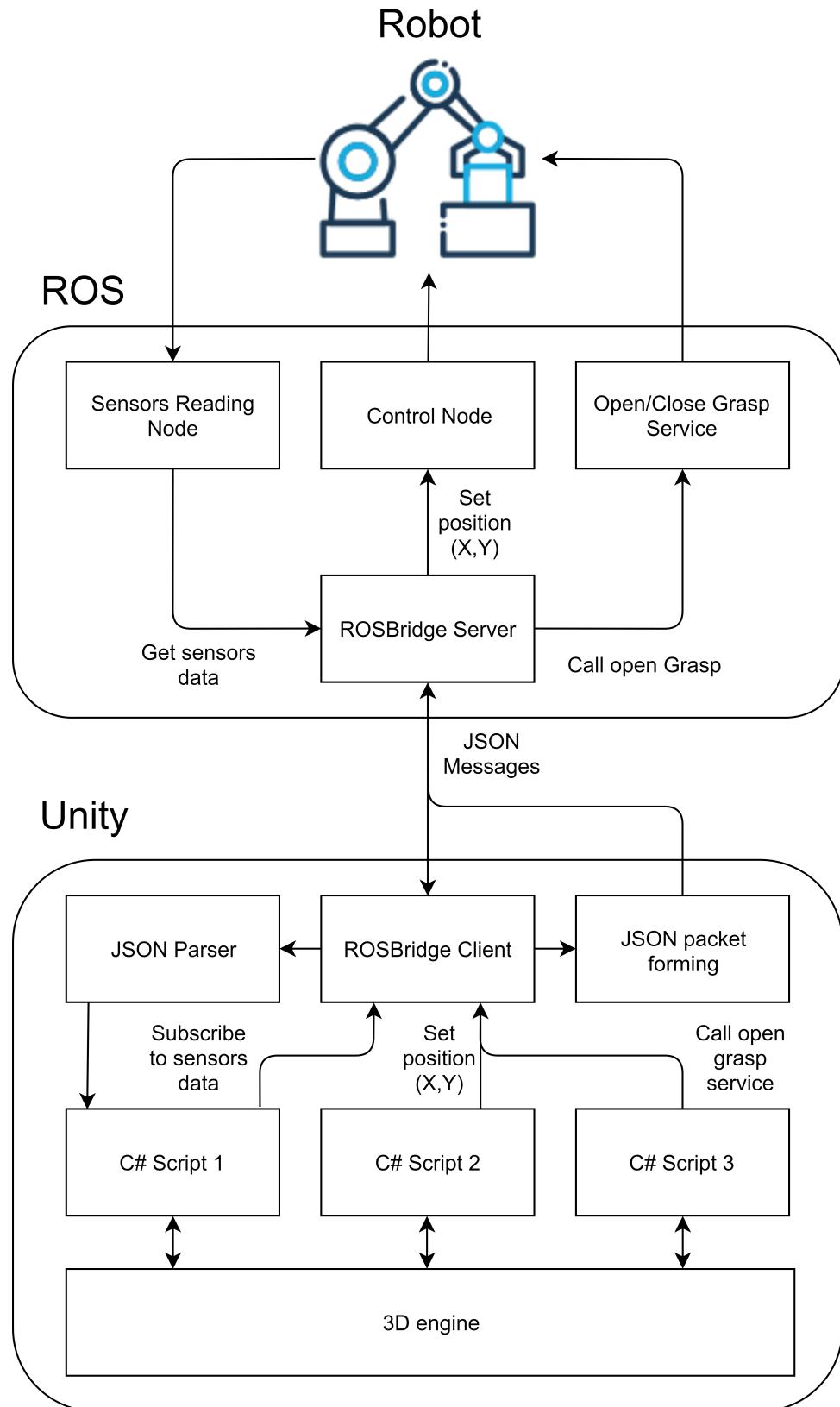


Figure 4.5: Example of connection between ROS and Unity via ROSBridge

With the ability to exchange data with ROS, implementations of the basic assumption

tions have started. Initially, the coordinate systems of the robot and goggles had to be synchronized so that the objects oriented in the virtual space would correspond to reality. For this purpose, the aforementioned Vuforia library was used. Created marker need to be placed in exact same position in front of robotic arm as it was set in Unity 3D environment. Thanks to that all objects will appear just in a right spot.

Next task was to create a 3D cube and bind its position with the position of the robot effector taken from ROS. This test allowed to observe that the coordinate axes of the robot are pointing in other directions than Unity one's. To facilitate the later work, another script dealing with bidirectional transformation was written.

After the application was able to receive a position from the robot, a part of the code allowing its setting was written. In this case gesture recognition module from HoloToolkit library was used. It allows to capture event when particular gesture is occurring and extract from it information about relative position of hand. Using this from the perspective of the code, we are able to add a hand translation vector to the position vector of the selected object which will appear as its shift in space. This allows to move previously created cube, then reads its position and at the end write it to robot. That part works very well. In Figure 4.6 screenshots from hololens with different positions set can be seen.



Figure 4.6: Setting robotic arm position by gesture

Last part of the task was to give ability to store set positions, draw path between them and at the end to be able to automatically replay a sequence of movements. For this purpose click gesture was used. When user set desired position by dragging followed by robot object then he can click on it. This event will instantiate a sphere at that place to visualize and store position. In addition, a path is drawn between the next steps to present the movement of the robot.

4.2.2 Mobile robot

In this part of research, connection between Unity and ROS was done in the same way like in case of robotic arm. The only difference between them was number of supported messages types. It was needed to be extended to be able to receive sensors data.

This task required one basic assumption, that is, the robot had to be able to locate itself. Unfortunately, the camera built into the Hololens does not have a very high resolution so that it would not be able to provide location information in sufficient accuracy. Therefore, the marker on the robot only serves to synchronize the coordinate systems with goggles. Again like in robotic arm experiment it is detected by Vuforia Engine.

Robot used in this part of research is called TurtleBot3 Waffle [20]. This is an open-source, open-hardware modular platform which allows to quickly test new ideas and algorithms. Its heart is Raspberry Pi 3 with ROS installed on it. Two digital servos from dynamixel are used to drive the wheels. It also has LiDAR installed. Picture of robot can be found in Figure 4.7.

First, the task related to the visualization of sensory data was carried out. Based on the fact that reading information from the robot had already been developed, it was

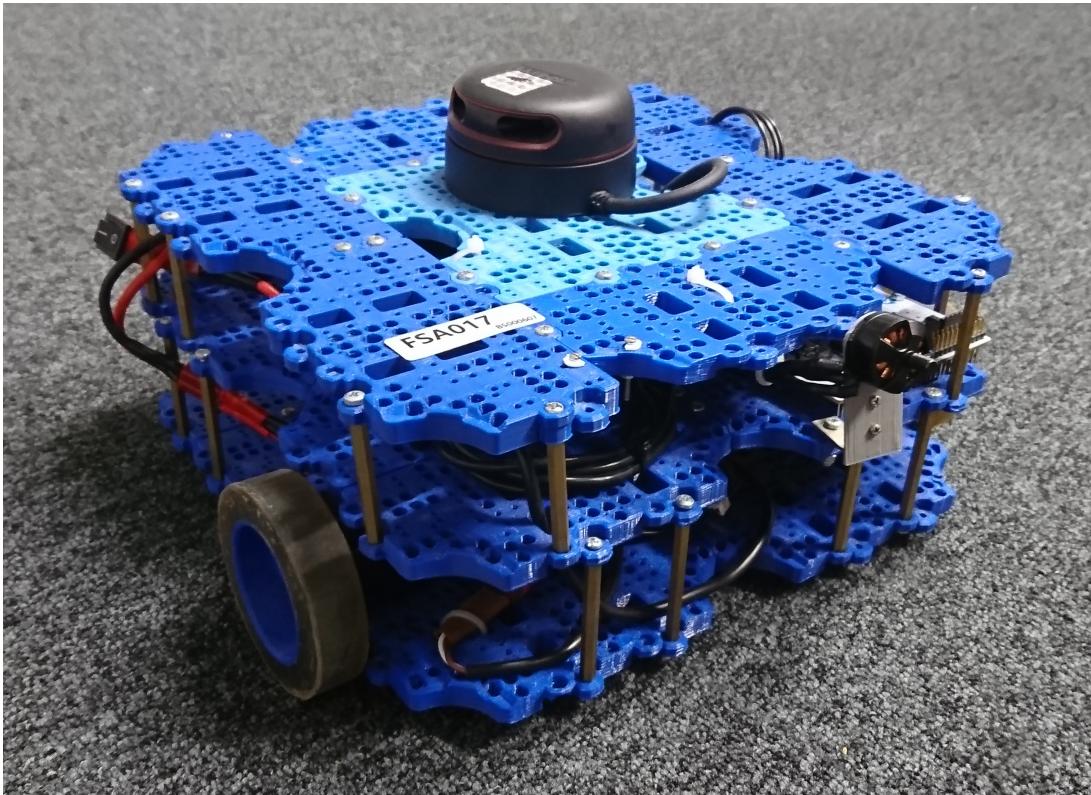


Figure 4.7: Turtlebot3 Waffle robot

quite easy to get to it. When connection to ROSBridge is established, information about subscription need to be send to the server to obtain data from a specific topic. After performing this operation, values update will be received by given callback function. From there they go to the script which was written to generate small spheres on specific positions corresponding to the read distance data provided by robots LiDAR sensor. This gives visual feedback what objects are seen by it and which are not. In example at Figure 4.8 there can be seen erroneous points caused by LiDAR beam reflections on glass doors.

The second task was quite similar to what was done with the robot arm. Prepared application allowed to mark points on the floor to which the robot has to go. The C# script created in Unity uses the click gesture to retrieve the cursor position indicating the place the user is looking at. Then a sphere is instantiated at that place and as long as the gesture is not released, it is possible to change the orientation using the hand movement. The operation is visualized by means of a rotating arrow attached to the created sphere. After completing the gesture, the position and orientation of the selected point from the goggles coordinates to the coordinates of the robot are converted. Finally, a JSON query is created from the obtained data, which is sent via ROSBridge. On robot side there is called move_base action. This is a node which is linking together local and global planner. Thanks to it, on the basis of data from sensors and odometry, the robot is able to reach the point by solving global navigation task. In Figure ?? there can be seen screenshots from hololens showing placed virtual marker and robot approaching to it.

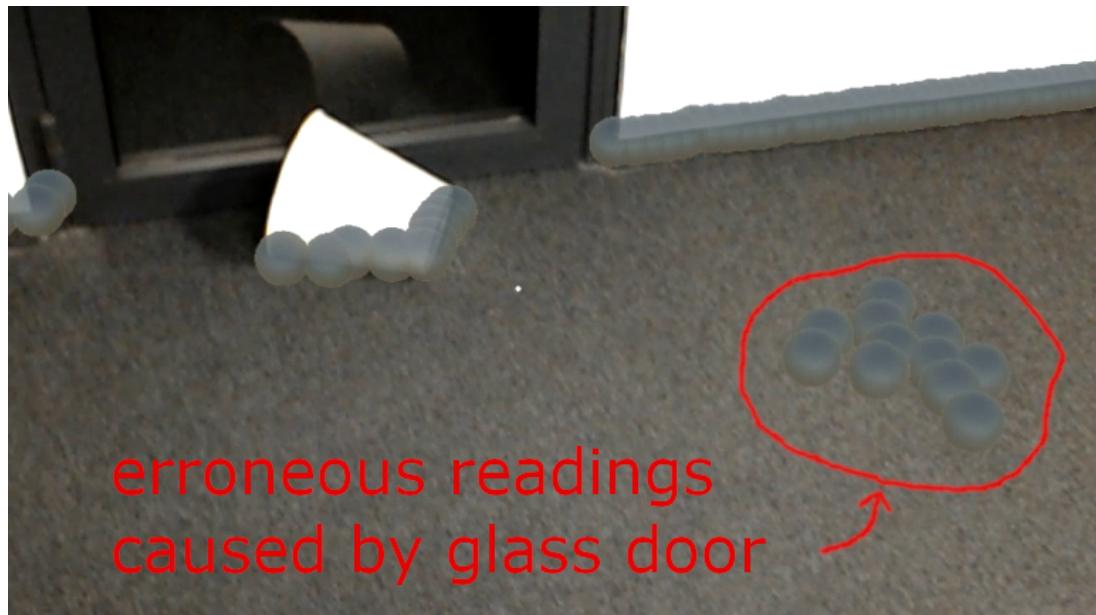


Figure 4.8: Visualization of LiDAR sensor data, with visible erroneous points

Chapter 5

Summary

The ways of using augmented reality in robotics presented in this thesis are certainly not exhausting this topic. Due to the relatively early stage of development of this technology, only simple cases were chosen. Nevertheless, all selected tasks have been completed. In each case, the prepared interface allowed to perform the required actions, but usually the precision of movements was not satisfactory. It was influenced by the problem of keeping hand in one place and probably by inaccurate goggles positioning. It is possible that improving some aspects of the proposed user interface would improve this issue.

Returning, however, to the advantages, certainly a big plus was the ability to observe working sensors, so that the user had constant information what does the robot see and what does not at the moment. This allowed several times for faster response in the event of an error in implementation. In addition, tests of the proposed UI were carried out on persons not related to robotics. Each of them spoke very positively about it and was able to operate given unit after a small instructor.

Augmented Reality is a super fast growing technology. It allows creating more intuitive user interfaces due to the ability to generate spatial objects. Thanks to its use, it is possible to facilitate a lot of production processes as well as increase work safety. Of course, the use of this technology is much wider than just industrial applications.

Unfortunately, devices that give the best results at the moment are too expensive for the average user and often too uncomfortable to use on a daily basis. Another reason for the low popularity of this technology beyond the price and comfort is still insufficient computing power to generate decent quality materials and problems with the projection of the image with a wide viewing angle. In addition, often the detection of gestures is not as precise as expected by the user. However, thanks to the increasing use of this technology in the industry, newer and newer solutions are introduced which helps to improve all aspects of this technology.

Who knows, it's possible that in the near future these devices will be as popular as today's smartphones and maybe even take over their role.

Bibliography

- [1] ALVAR official site. <http://virtual.vtt.fi/virtual/proj2/multimedia/alvar/>. Accessed: 29-01-2019.
- [2] ARCore official site. <https://developers.google.com/ar/>. Accessed: 29-01-2019.
- [3] ARKit official site. <https://developer.apple.com/arkit/>. Accessed: 29-01-2019.
- [4] ARVIND SANJEEV lumen. <http://arvindsanjeev.com/lumen.html>. Accessed: 29-01-2019.
- [5] Continental head-up displays. <http://continental-head-up-display.com/>. Accessed: 29-01-2019.
- [6] DAQRI smart glasses. <https://daqri.com/products/smart-glasses/>. Accessed: 29-01-2019.
- [7] Dobot official site. <https://www.dobot.cc/event/new-update-to-dobot-magician-comparison-between-v1-and-v2.html>. Accessed: 29-01-2019.
- [8] Dwm1000 module. <https://www.decawave.com/product/dwm1000-module/>. Accessed: 29-01-2019.
- [9] Epson moverio pro bt-2200. <https://www.epson.pl/products/see-through-mobile-viewer/moverio-pro-bt-2200>. Accessed: 29-01-2019.
- [10] Google glass. <https://www.x.company/glass/>. Accessed: 29-01-2019.
- [11] Google Glass uses and features. <https://sites.google.com/site/g00g13g1a55/google-glass>. Accessed: 29-01-2019.
- [12] Kinect – opensource [news]creativeapplications.net. <https://doomicile.de/story/kinect-opensource-newscreativeapplicationsnet>. Accessed: 29-01-2019.
- [13] Light Guide Systems classic. <https://lightguidesys.com/products/light-guide-systems-classic/>. Accessed: 29-01-2019.
- [14] Magic Leap one. <https://www.magicleap.com/magic-leap-one>. Accessed: 29-01-2019.
- [15] Microsoft Hololens gestures. <https://docs.microsoft.com/en-us/windows/mixed-reality/gestures>. Accessed: 29-01-2019.
- [16] MS Hololens hardware accessories. <https://docs.microsoft.com/en-us/windows/mixed-reality/hardware-accessories>. Accessed: 29-01-2019.

- [17] MS Hololens hardware details. <https://docs.microsoft.com/en-us/windows/mixed-reality/hololens-hardware-details>. Accessed: 29-01-2019.
- [18] MS Hololens official site. <https://www.microsoft.com/en-us/hololens>. Accessed: 29-01-2019.
- [19] New motion capture equipment for digital arts students! <https://blogs.chapman.edu/dodge/2013/10/29/new-motion-capture-equipment-for-digital-arts-students/>. Accessed: 29-01-2019.
- [20] TurtleBot official site. <https://www.turtlebot.com/>. Accessed: 29-01-2019.
- [21] Vuforia Engine official site. <https://www.vuforia.com/engine.html>. Accessed: 29-01-2019.
- [22] A. Babinec, L. Jurišica, P. Hubinský, and F. Duchoň. Visual localization of mobile robot using artificial markers. *Procedia Engineering*, 96, 12 2014.
- [23] M. Field, D. Stirling, F. Naghdy, and Z. Pan. Motion capture in robotics review. In *2009 IEEE International Conference on Control and Automation*, pages 1697–1702, Dec 2009.
- [24] P. Fraga-Lamas, T. M. Fernández-Caramés, . Blanco-Novoa, and M. A. Vilar-Montesinos. A review on industrial augmented reality systems for the industry 4.0 shipyard. *IEEE Access*, 6:13358–13375, 2018.
- [25] Z. Materna, M. Kapinus, V. Beran, P. Smrz, and P. Zemcík. Interactive spatial augmented reality in collaborative robot programming: User experience evaluation. pages 80–87, 08 2018.
- [26] P. Milgram, H. Takemura, A. Utsumi, and F. Kishino. Augmented reality: A class of displays on the reality-virtuality continuum. *Telemanipulator and Telepresence Technologies*, 2351, 01 1994.
- [27] A. Millard, R. Redpath, A. M. Jewers, C. Arndt, R. Joyce, J. Hilder, L. Mcdaid, and D. M. Halliday. Ardebug: An augmented reality tool for analysing and debugging swarm robotic systems. *Frontiers in Robotics and AI*, 5:87, 07 2018.
- [28] D. Molyneaux, S. Izadi, D. Kim, O. Hilliges, S. Hodges, X. Cao, A. Butler, H. Gellersen, A. Butler, D. Molyneaux, O. Hilliges, S. Izadi, S. Hodges, and X. Cao. Interactive environment-aware handheld projectors for pervasive computing spaces. *Pervasive Computing*, pages 197–215, January 2012.
- [29] A. Reina, M. Salvaro, G. Francesca, L. Garattoni, C. Pincioli, M. Dorigo, and M. Birattari. Augmented reality for robots: Virtual sensing technology applied to a swarm of e-pucks. In *2015 NASA/ESA Conference on Adaptive Hardware and Systems (AHS)*, pages 1–6, June 2015.