

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, I-6

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	9
2.2	Products available on market	10
2.2.1	Hardware devices	10
2.2.2	Software solutions	11
3	Applications of Augmented Reality	15
3.1	ARDebug	15
3.2	Virtual sensing technology	15
3.3	Interactive Spatial AR in Collaborative Robot Programming	17
4	Research of the subject	19
4.1	Used technologies	19
4.1.1	3D Engine	19
4.1.2	Vuforia	20
4.1.3	Robotic Operating System	21
4.2	Test results	21
4.2.1	Robotic arm	21
4.2.2	Mobile robot	23
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 on home appliances robots. They all have more or less user friendly interface created to program or control them. In factories can be seen most often stationary or hand-held controllers and in consumer appliances, smart-phone almost every time is used. Problem is that, this kind of interaction is not natural for humans. For comparison, communication between two employees working together is mostly done by voice, gestures and sometime 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 are trying 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 are extending their perception. This is allowing to get better understanding what machines are doing or even see what they are "thinking".

1.1 Purpose and scope of work

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

Research part of this thesis will try to present simple examples of implementation AR in robotic applications. The topic will cover the issue of planning movement of robotic arm and also controlling and presenting data from mobile robot. This should give more or less understanding what this technology is capable 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 will try to trick him 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" [22] was created. Its graphical representation is shown on figure 2.1.

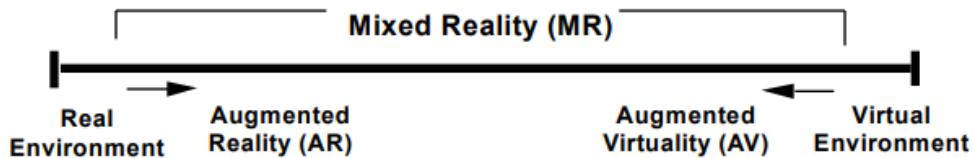


Figure 2.1: Reality-Virtuality (RV) Continuum [22]

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 other hand on the most right side there is Virtual Reality with fully 3D generated world that could even in example not hold known by us laws of physics. Between two of those is everything with is mixing one part with another. Depending on what balance is, then we will talk about Augmented Reality or Augmented Virtuality. For example when there operator is getting some simulated cues to augment his natural feedback then it is 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 are working at green screen and in television they appear in 3D generated studios. As this paper will only discuss topic of using Augmented Reality in robotics applications that is why technical aspects of Augmented Virtuality or Virtual Reality will be not considered.

2.1 Technology overview

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



Figure 2.2: Simplified AR pipeline [20]

This section will show different types of technologies with where used at implementation of individual elements of this process.

2.1.1 Types of image projection

There is several way of displaying virtual objects in our environment. Depending on its type they could be more or less immersing for the user.

The most popular and the simplest types of projection is video-mixing. It involves the usage of devices equipped with a camera and standard display to present AR content. In most cases smartphones or tablets are used because they have built-in every needed component and they are very portable. Figure 2.3 present 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.

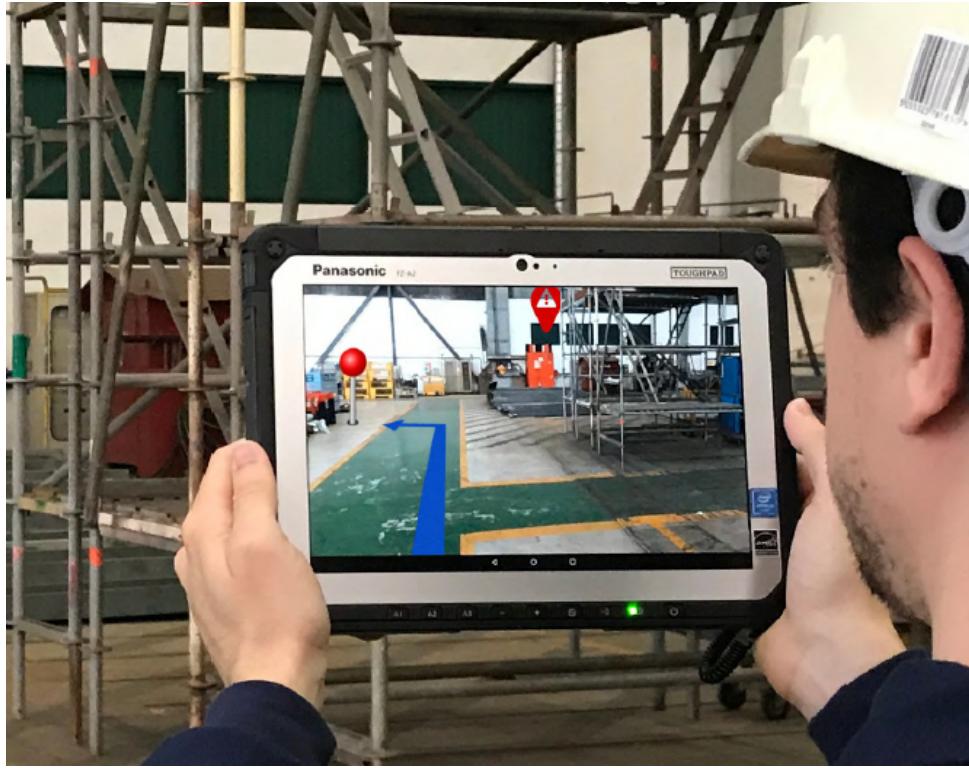
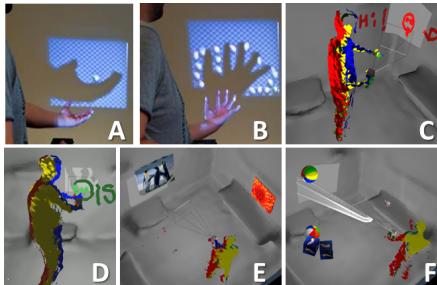


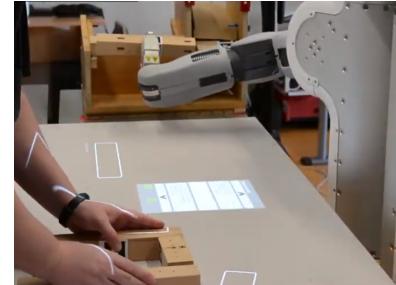
Figure 2.3: Worker using tablet as video-mixing device [20]

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 on two way. We could have hand-held device with will work similar to flashlight or stationary mounted projector. In 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 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 worker which item it will pick-up or where it will place it on shared workspace. Presented types of projection could be seen on Figure 2.4.



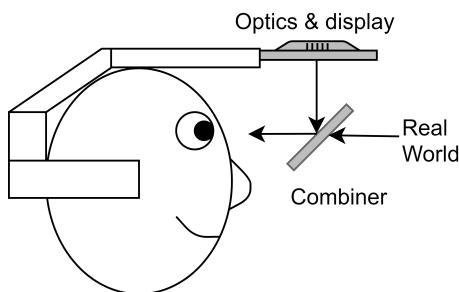
(a) Augmented projector [24]



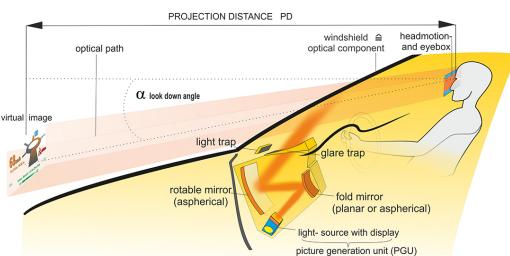
(b) Man and robot coworking [21]

Figure 2.4: Examples of spatial projection

Next AR display technology which will be presented is optical see-through. Its principle of operation boils down to projecting images on partially reflecting surface. This allow to combine real world view with generated graphics which will appear as floating in space holograms. 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 could be shown some step by step instructions or in case operating with robots, live sensors data. Example from image 2.5b show 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 car see and what it intends to do.



(a) Head-Mounted Display



(b) Head-Up Display [2]

Figure 2.5: Examples of optical see-through projections

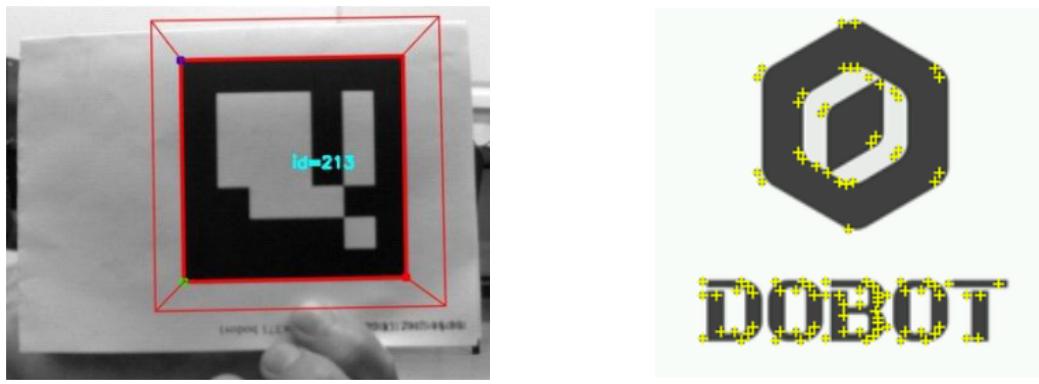
2.1.2 Positioning and location

Only small part of AR devices is presenting data that is not depending on real environment obstacles. These include basic versions of HUD and smartglasses. Rest of them need somehow localize itself or detect position of the user. In this section various types of positioning techniques will be presented.

Marker-based localization is 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.

Most trivial example is when our marker have square shape and high contrast. In that case algorithm need only know what is length of 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. This allow to store some information inside and also help to determinate from which side of square camera is looking. Figure 2.6a show detected marker with read associated to it ID number. Pros of this approach is quite low requirement for computational power and easy way to generate many unique ID's. The disadvantage is that the user must print these images and arrange them in their chosen locations.

There is also possibility to track normal images or even 3D objects. In this case there are used algorithms with are 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 [18] (b) Image with marked extracted features

Figure 2.6: Example of markers used for localization purposes

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

Optical tracking of passive markers [19] 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 sub-millimeter precision. To avoid situation where marker is occluded by some obstacles localized object could be equipped with redundant ones. 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 range by using active markers which are light sources but this cause another problems with 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 with are providing not only image but also give information about distance from objects. This allow to not only get position of tracked

device but also could give feedback about scanned environment. This kind of systems generally have operation distance from 0.5 to 8 meters. Precision of positioning it 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 compare to optical ones but have huge advantage in the form of being able to track device without direct visibility. Also range and refresh rate of such systems is significantly larger than optical ones.

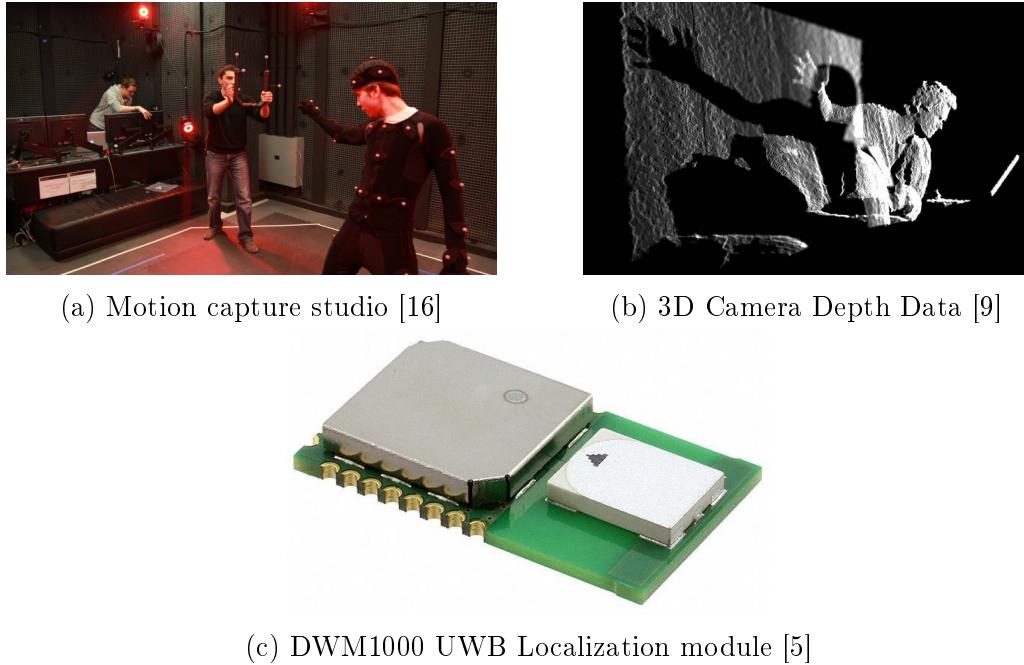


Figure 2.7: External localization systems

Last presented type of positioning is Simultaneous Localization And Mapping (SLAM). This is technique with is using 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 with makes it almost limitless in terms of working area. Also additional gain is that the obtained data can be used to classify objects located in the environment. Unfortunately, such calculations takes huge amount of processing power so device need more expensive components to be able run in real-time. At Figure 2.8 could be seen 3D map of room created by using depth sensor and SLAM algorithm.

2.1.3 User control and interaction

Augmented Reality devices requires different approach to how user should use them. As the worker is no longer limited by his stationary desk there is need to change way of interaction. This section will discuss most common used methods.

Simplest HMD devices, where in the field of view we had only displayed text or images there is no need to add special types of interaction. In most cases on frame of device are placed buttons or small touchpad (Figure 2.9a) which allow user to control displayed content. This approach allows to reduce production costs so that it could have lower price tag and be able to get to more customers.

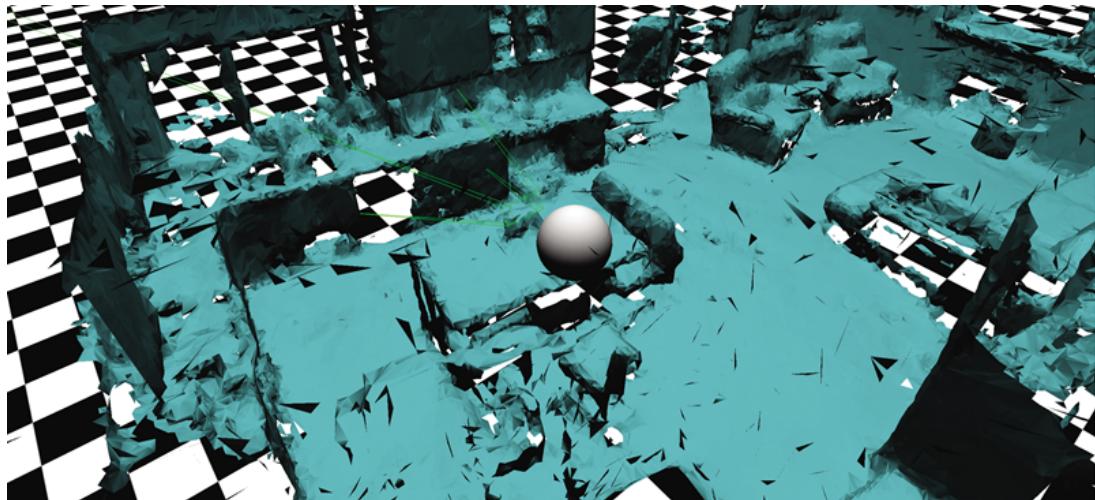


Figure 2.8: Room mapped using SLAM algorithm with usage of Microsoft Hololens

When device is providing ability to observe 3D objects in space then user need some controllers to be able to operate them. On the market most popular ones are using optical tracking (Figure 2.9b) but there are also those that uses electromagnetic fields. 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 allow 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 consume a lot of computing power so it is reserved only for the most expensive devices.

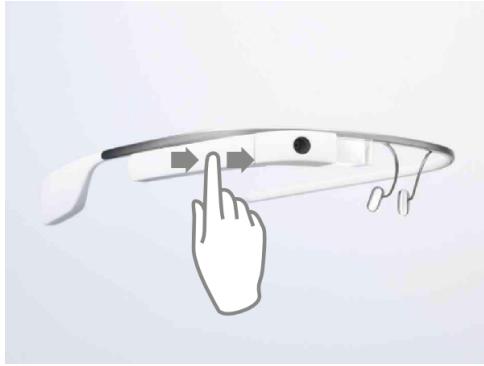
The last type of interaction with will be discussed is voice control. It is a very intricate topic due to diversity of humans dialects and accents. With current technology embedded computers can interpret only single words or predefined sentences however, using the resources of cloud services it is possible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

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 on the market. They use a variety of image display and interaction technologies, and develop newer and newer solutions. This section will show few chosen AR devices and software's.

2.2.1 Hardware devices

The devices presented at table 2.1 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



(a) Touchpad at Google Glass [8]



(b) Mixed Reality Controllers [13]



(c) Using gesture with Microsoft Hololens [12]

fulfill specific case. However, some devices can be bought in normal sales, so it is not a closed market for a regular customer.

2.2.2 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 2.2 shows the collected information about several of them.

Device name	Type of projection	Hardware	Sensors	Connectivity	Battery (life time)	Mechanical parameters	User input	Operating system	Capabilities	Price
Microsoft Hololens[14]	Optical see-through 1920x1200 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	4300 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 Mapping of environment Teleoperation	\$4500
Epson MOVERIO PRO BT-2000/2200[6]	Optical see-through 960x540 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: 276-300 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[3]	Optical see-through 1300x768 px per eye 44 degrees view angle	6th Intel® Core™ i7 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: 335 g Compute Pack: 406 g Adjustable head band	Gesture input Voice support	Custom solution	Mapping of environment Teleoperation 6-DOF tracking	Unknown
ARVIND SANIEEV LUMEN[1]	Projection based 640x360px screen size: 10"-11.2"	Broadcom BCM7285 1GHz 512 MB RAM 32 GB external microSD	Color Camera Depth Sensor Camera	Wi-Fi 802.11b/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[0]	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[11]	Optical see-through 1280x960 px per eye 50 degrees view angle	CP U: NVIDIA® Parker GP U: 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[7]	Optical see-through 640x360 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 Touchpad Mobile app	Android™ 4.4	Teleoperation Assistant Task guidance	\$1,500

Table 2.1: List of selected AR goggles models

2.2. Products available on market

Year	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010
GPD in billions	235	225 bn	223 bn	323	423	523	624	725	826	924	1022

Table 2.2: Placeholder

Chapter 3

Applications of Augmented Reality

This part of thesis will be presenting some founded on internet publications about using AR in robotics applications. It should give a pretty good look at what this technology is created for and suggest what areas have not yet been researched.

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 [23] created by scientists from York Robotics Laboratory. The tool presented by them could help to easier find bugs in implemented algorithms by giving real-time visual feedback basing on each robot state.

The configuration of the environment requires the use of a locating system. Researchers for this purposes used an 5 MP camera which was placed 2.5 m above 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 in this way was packed in the JSON structure with is accepted by ARDebug application. Authors deliberately chose this format to allow easy integration of other location methods.

Another element required by the application is data of the status of each robot. They must also 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 at figure 3.1.

3.2 Virtual sensing technology

Another discussed 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 [25] of scientists from Universit Libre de Bruxelles was examined. In this paper they describe created by them system ARGoS with allow to simulate sensors input without having installed real ones. 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.

Like in previous system there is also need of reading robots positions. It was realized in the same way, that is, by using a camera suspended above the working area and markers

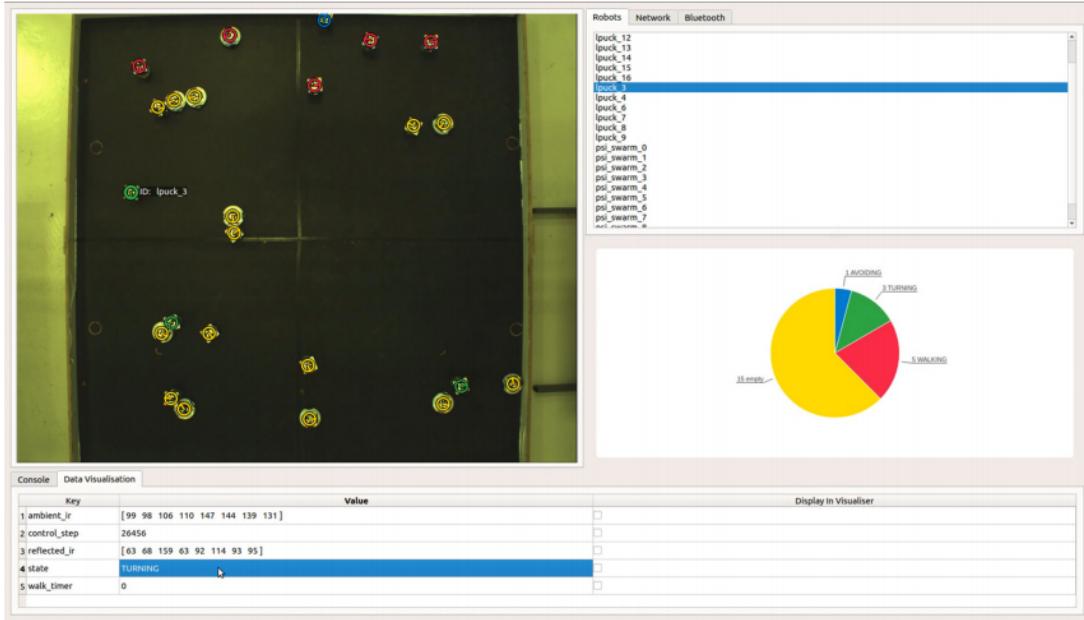


Figure 3.1: The ARDebug user interface [23]

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 could be seen na figure 3.2.

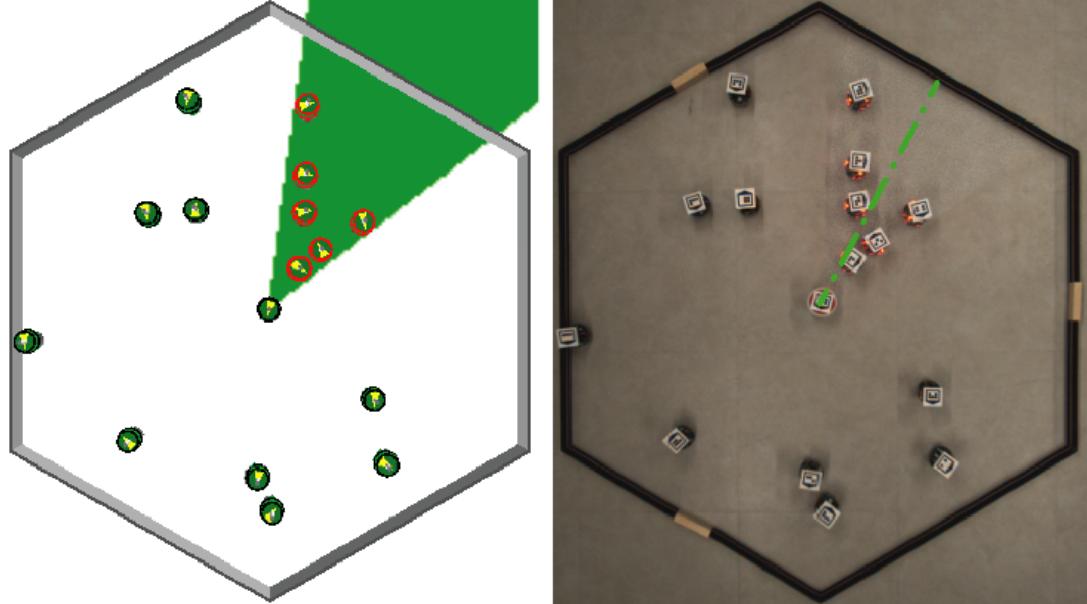


Figure 3.2: The ARGoS sensor simulator [25]

3.3 Interactive Spatial AR in Collaborative Robot Programming

Scientists from Brno University of Technology decided to explore the subject of human cooperation with the machine. In their paper [21] described was concept of utilization of shared workspace as a interactive space with could be used to visualize and control movements of collaborative robot. The goal was to create 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.

Researchers to test different solutions have created a 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 could be seen on figure 3.3.



Figure 3.3: Setup of collaborative workshop desk [21]

Prepared elements changes shared workspace into interactive display. Thanks to this, the 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 finds out

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

Chapter 4

Research of the subject

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 an user interface that would allow to remotely control the position of the robotic arm using gestures. Thanks to this, working from a safe distance, the operator is able to set a trajectory of motion along the points, 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 reading in user environment could help with debugging some hardware or software issues. Also being able to see waypoints of robot on the floor could help to check if it will not 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 could be seen on figure 4.1. Technical parameters was presented earlier in the table 2.1.



Figure 4.1: Picture presenting Microsoft Hololens AR Googles [15]

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. Also allow to have more control on code execution, due to possibility of writing backend in C++. Unfortunately it do 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 need to be created C# script file.

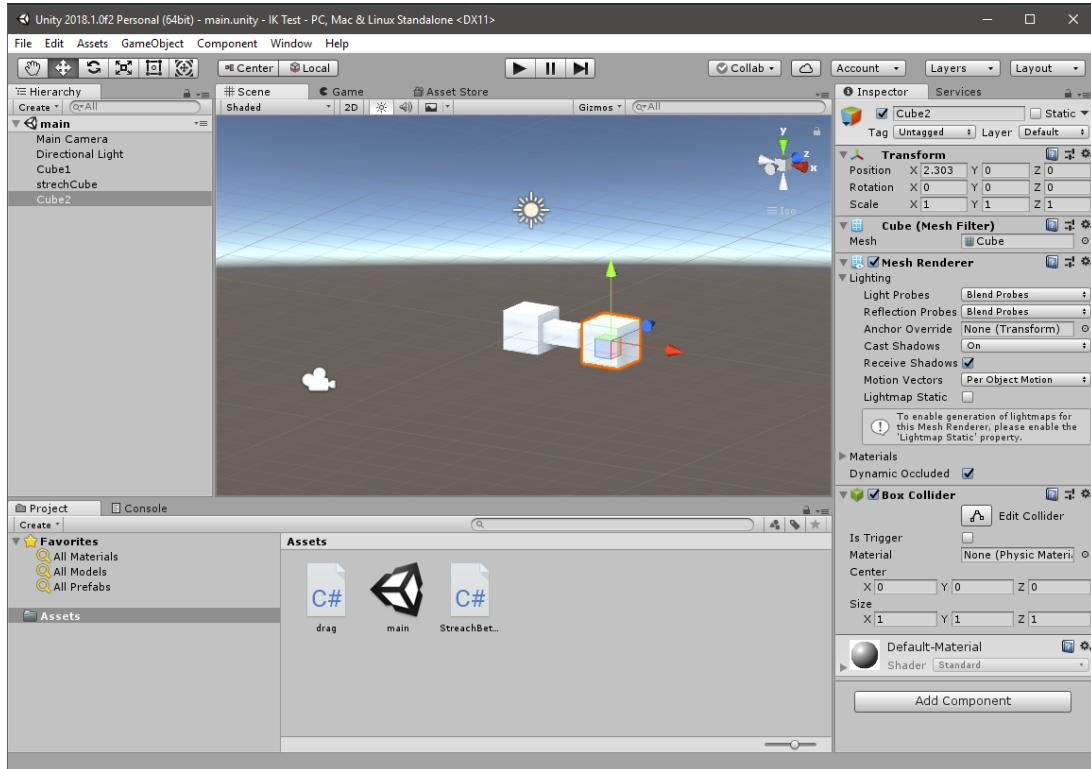


Figure 4.2: Unity Editor main window

Developing applications for Microsoft Hololens in the Unity Engine has additional advantages in terms of the availability of the HoloToolkit library. It allows you 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 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 2.2

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. At figure 2.6b could 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 with provide 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 an 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.

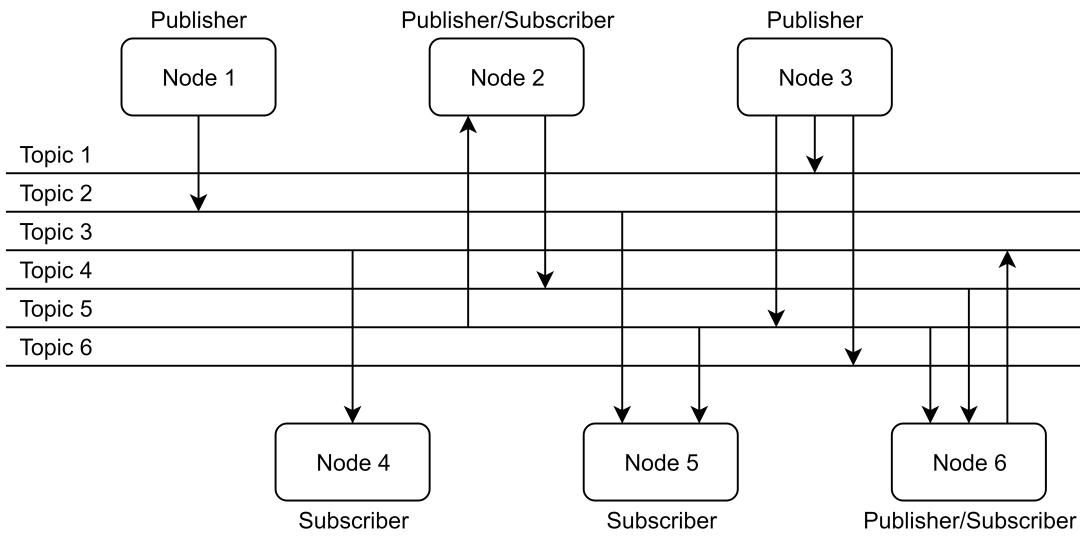


Figure 4.3: ROS publish/subscribe mechanism

Another great feature is also ability to record and playback messages. As the data in 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 [4]. This is 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 homeing action. It could be seen at figure 4.4.



Figure 4.4: Dobot Magician robotic arm [4]

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.

With the ability to exchange data with ROS, implementations of the basic assumptions 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 setted 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 application was able to receive an 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 allow to move previously created cube, then reads its position and at the end write it to robot. That part works very well. On figure 4.5 could be seen screenshots from hololens with different positions setted.



Figure 4.5: Setting robotic arm position by gesture

Last part of the task was to give ability to store setted positions, draw path between them and at the end be able to automatically replay sequence of movements. For this purpose an click gesture was used. When user set desired position by dragging followed by robot object then he could 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 need to be extend 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 [17]. This is an open-source, open-hardware modular platform which allow to quickly test new ideas and algorithms. His 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 could be find at figure 4.6.

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 quite easy to get to it. After connection is established and data is acquired script which was written is generating small spheres and pushes them away from the center of LiDAR at a

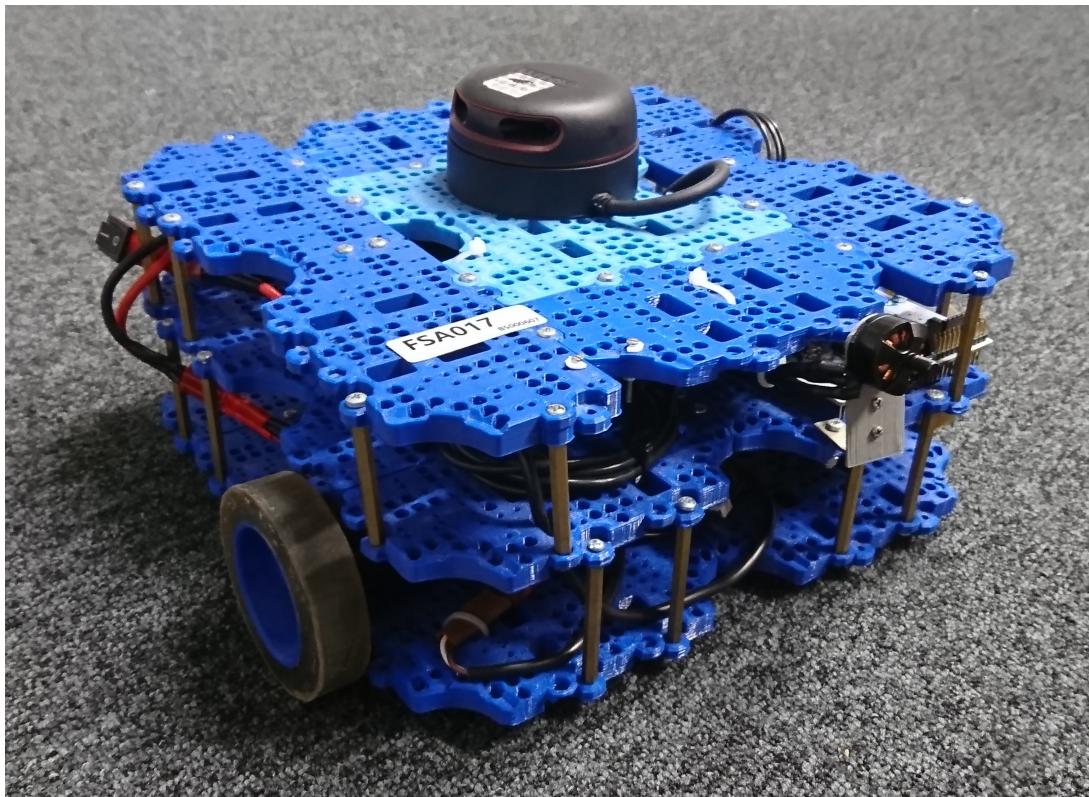


Figure 4.6: Turtlebot3 Waffle robot

distance read from the ROS. This give us visual feedback which objects could be seen by given sensor. In this case LiDAR beam could not detect glass doors (Figure 4.7).

Figure 4.7: Visualization of LiDAR sensor data

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. From Unity side it was quite easy task. The more complex work was done on robot, because he need to calculate path to be able to reach that point. Fortunately, the built-in ROS libraries came to the rescue and they are able to do this job quite well. The results of this work could be seen at figure ??.

Chapter 5

Summary

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

- [16] 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.
- [17] TurtleBot official site. <https://www.turtlebot.com/>. Accessed: 29-01-2019.
- [18] A. Babinec, L. Jurišica, P. Hubinský, F. Duchoň. Visual localization of mobile robot using artificial markers. *Procedia Engineering*, 96, 12 2014.
- [19] M. Field, D. Stirling, F. Naghdly, Z. Pan. Motion capture in robotics review. *2009 IEEE International Conference on Control and Automation*, strony 1697–1702, Dec 2009.
- [20] P. Fraga-Lamas, T. M. Fernández-Caramés, . Blanco-Novoa, M. A. Vilar-Montesinos. A review on industrial augmented reality systems for the industry 4.0 shipyard. *IEEE Access*, 6:13358–13375, 2018.
- [21] Z. Materna, M. Kapinus, V. Beran, P. Smrz, P. Zemcík. Interactive spatial augmented reality in collaborative robot programming: User experience evaluation. strony 80–87, 08 2018.
- [22] P. Milgram, H. Takemura, A. Utsumi, F. Kishino. Augmented reality: A class of displays on the reality-virtuality continuum. *Telemanipulator and Telepresence Technologies*, 2351, 01 1994.
- [23] A. Millard, R. Redpath, A. M. Jewers, C. Arndt, R. Joyce, J. Hilder, L. Mcdaid, D. M. Halliday. Ardebug: An augmented reality tool for analysing and debugging swarm robotic systems. *Frontiers in Robotics and AI*, 5:87, 07 2018.
- [24] 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, X. Cao. Interactive environment-aware handheld projectors for pervasive computing spaces. *Pervasive Computing*, strony 197–215, January 2012.
- [25] A. Reina, M. Salvato, G. Francesca, L. Garattoni, C. Pincioli, M. Dorigo, M. Birattari. Augmented reality for robots: Virtual sensing technology applied to a swarm of e-pucks. *2015 NASA/ESA Conference on Adaptive Hardware and Systems (AHS)*, strony 1–6, June 2015.