UNIVERSITY OF TARTU

Faculty of Science and Technology

Institute of Technology

Igor Rybalskii

Augmented reality (AR) for enabling human-robot collaboration with ROS robots

Master's Thesis (30 ECTS)

Curriculum Robotics and Computer Engineering

Supervisors:

Associate professor Karl Kruusamäe

Tartu 2022

Abstract

Augmented reality (AR) for enabling human-robot collaboration with ROS robots

With current industrialization trends on the reintroduction of human into the manufacturing process and development of augmented reality I propose the interface, which uses Augmented reality to allow the operator to interact with robotic systems, such as manipulators and mobile robots. 2 interfaces were created: one for manipulators and one for mobile robots. Both of them are developed to work on Microsoft Hololens 2 augmented reality glasses and robots, running ROS. Interface for manipulators allows user to control the robot by sending end-effector goals and previewing the goal joint states of the manipulator. Interface for mobile robots allows the user to send navigation goals and preview the robot's movement trajectory. Interfaces were developed using Unity game engine. Developed interfaces were tested with UR5e manipulator and Robotont mobile robot.

Keywords: human-robot interaction, augmented reality, Microsoft Hololens 2, ROS, Unity 3D game engine

CERCS: T120 - Systems engineering, computer technology, T125 - Automation, robotics, control engineering

Lühikokkuvõte

Liitreaalsus inimese ja roboti koostöö võimaldamiseks ROS-i robotitega

Seoses praeguste industrialiseerimise arengusuundadega, mis käsitlevad liitreaalsuse arendamist ja inimese tootmisprotsessi taastoomist, mina pakun välja liidese, mis võimaldab operaatoril vastastikku toimima robotsüsteemidega, nagu manipulaatorid ja mobiilsed robotid, rakendades liitreaalsust. Loodi kaks liidest: üks manipulaatorite ja üks mobiilsetele robotite jaoks. Mõlemad on välja töötatud jooksma Microsoft Hololens 2 liitreaalsusprillide ja robotitega, mis käivitavad ROS-i. Manipulaatorite liides võimaldab kasutajal robotit juhtida, saates lõpp-efektori sihtmärke ja kuvades manipulaatori liigeste sihtmärgiolekutest

eelvadet. Mobiilsete robotite liides võimaldab kasutajal saata navigeerimissihtmärke ja vaadata roboti liikumistrajektoori. Liidesed töötati välja Unity mängumootori peal. Ilmunud liideseid testiti UR5e manipulaatori ja Robotont mobiilse robotiga.

Võtmesõnad: inimene ja roboti suhtlus, liitreaalsus, Microsoft Hololens 2, ROS, Unity 3D mängumootori

CERCS: T120 - Süsteemitehnoloogia, arvutitehnoloogia, T125 - Automatiseerimine, robootika, juhtimistehnika

Table of contents

Abstract: 1

Lühikokkuvõte: 1

Table of contents 2

1. Abbreviations 3

2. Introduction 4

3. Literature review 5

3.1 Augmented reality (AR) 5

3.1.1 Types of AR 5

3.1.2 AR devices 9

3.2 Application in robotics 10

3.3 Tools for integrating AR and robotics 16

3.3.1 ROS 16

3.3.2 Unity 17

3.3.3 ROS-Unity communication 17

ROS# (ros-sharp) 17

MirrorLabs 18

Unity-Robotics-Hub 18

4. Technical requirements 19

4.1. Functional requirements 19

4.2. Non-functional requirements 19

5. Design 20

5.1 AR interface for commanding robot manipulators 20

5.2 AR interface for commanding mobile robots 22

6. Results 25

6.1 Manipulator application 25

6.2 Mobile robot application 25

7. Summary 27

8. Acknowledgements 28

9. Bibliography 29

NON-EXCLUSIVE LICENCE TO REPRODUCE THESIS AND MAKE THESIS PUBLIC 32

1. Abbreviations

AR - Augmented Reality

HMD - Head Mounted Display

IMU - Inertial Measurement Unit

ROS - Robot Operating System

HUD - Heads-Up Display

UAV - Unmanned Aerial Vehicle

HRI - Human-Robot Interaction

MR - Mobile Robot

XR - eXtended Reality

URDF - Universal Robot Description Format

MRTK - Mixed Reality ToolKit

UWP - Universal Windows Platform

QR code - Quick Response code

TCP - Transmission Control Protocol

2. Introduction

Currently industrialization going through a new phase, which is named Industry 5.0. Previous phase, Industry 4.0, aimed at further automation of manufacturing through cyber-physical systems, the Internet of Things, cloud computing and excluding people from physical presence during the manufacturing process [1]. Industry 5.0 strives to shift focus back to people and build manufacturing processes around human beings. This means that one of the goals of Industry 5.0 is the development of human-machine interaction tools [2].

Another technological trend is Augmented Reality, which enhances the perception of the physical world by introducing digital, usually visual, information, which is perception-synchronized with physical objects and places around the user [3], [4].

In the context of these 2 trends, I propose an Augmented Reality interface, which would allow the operator to interact with robotics systems and perceive relative information about the robot.

3. Literature review

This chapter will cover the topics of augmented reality and devices, which can be used to see it; how AR can be applied in the field of robotics and development tools, which can be used to develop AR interfaces for robotics.

3.1 Augmented reality (AR)

According to IEEE, augmented reality is the perception of digital, usually visual, information, which is perception-synchronized with physical objects and places around the user [3].

3.1.1 Types of AR

There are multiple ways to experience AR: by using head-mounted displays (HMDs), handheld devices or projections on the physical objects.

HMDs [5]–[7] (Fig. 1) in most cases use transparent displays to overlay additional information onto real objects and use integrated cameras, laserscans, and Inertial Measurement Unit (IMU) to keep track of the physical environment [5], [6]. There are also examples of VR devices, which can operate in AR mode [8] (Fig. 2).

Figure 1: Example of a user interacting with the virtual environment in the real world.[9]

Figure 2: AR functionality of HTC Vive Pro VR headset. [8]

Being head-mounted, these types of devices allow users to always see digital information in front of their eyes with a field of view up to 50° [6] and potentially interact with the presented information. On the other hand, these solutions can be bulky and bigger in size, compared to ordinary glasses. If HMD is not equipped with an onboard computer, these devices will require a reliable connection to the computer, which limits the user's mobility.

Handheld devices, such as modern smartphones have enough performance to be used for AR - one of the more famous examples is the Pokemon Go (Fig. 3) game which introduced the concept of AR to the general audiences back in 2016 [10], [11].

Figure 3: Pokemon GO using the AR capability of the phone.[12]

AR apps on smartphones use the same type of sensors as in HMDs: cameras, lasers, and IMU. This type of AR is the most accessible to ordinary people, thus leading to the development of apps, which provide daily life use cases, such as AR rulers [13] or tools to preview furniture in physical rooms [14], [15].

Projection-based AR, as the name suggests, uses projectors to overlay necessary information onto real objects. It can be used in manufacturing to help workers [16] (Fig. 4) or warn the people about potential dangers or intentions of other actors [17], [18].

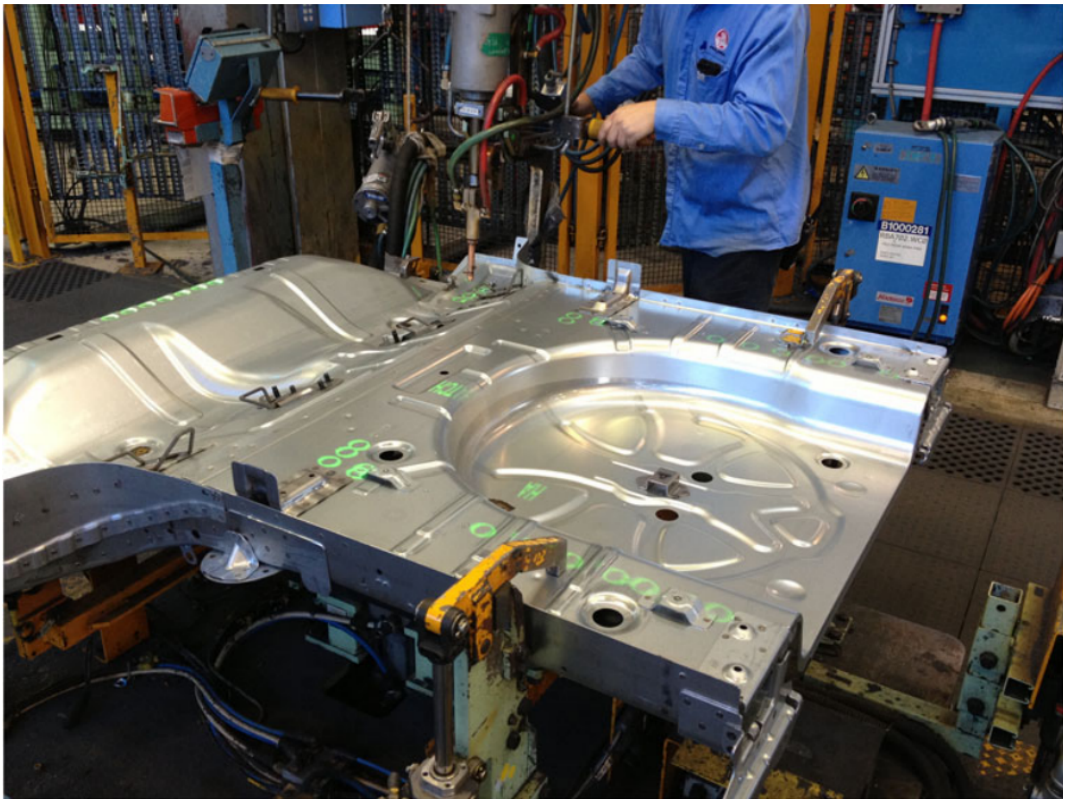
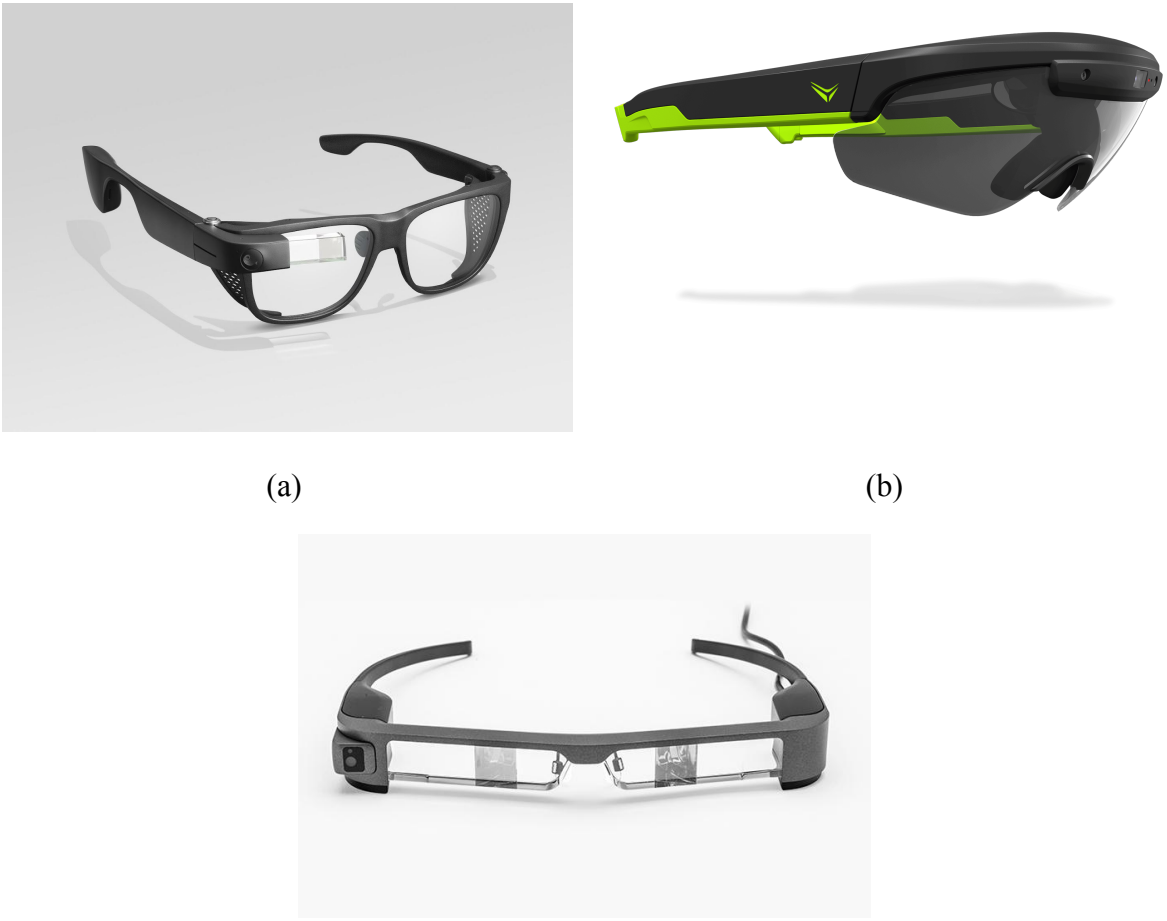


Figure 4: Usage of projections in manufacturing. [16]

3.1.2 AR devices

Quite possibly the most well-known AR device is the Google Glass (Fig. 5 a). It comes with a touchscreen, camera and a miniature display to overlay data [19]. The small form factor results in only 2 ways to operate these glasses: voice commands and a touchscreen on the side. The display itself doesn't overlay data according to the real environment and acts like a Heads-Up Display (HUD) which just overlays the information on the screen [20]. There are multiple similar portable AR glasses, with the same type of control and role of glasses as HUD. Examples can be Evesight Raptor (Fig. 5 b) for cyclists [7], Epson Moverio BT-300 (Fig. 5 c)for drone control [21]



(a) (b)

(c)

Figure 5 a: Google Glass Enterprise Edition 2 [22], b: Eversight Raptor [7], c: Epson Moverio BT-300 [21].

Microsoft Hololens 2 (Fig. 6) is one of the more advanced AR devices, which uses a combination of cameras and IMU to track its position in 3D space and map the environment.

It can track hand movement and gestures, such as pinching, clicking, and changing the size of an object. On-board computer allow these glasses to remain portable. [5]

Figure 6: Hololens 2

3.2 Application in robotics

AR is used to project digital data into a real environment. Robotics is a field, where an operator is required to work on a lot of different information, such as maps, laserscans, and robots state. Representing this information in a more localized way would allow decreasing operators' mental work, as they would not need to connect the virtual and real environments with each other.

In [23] AR is used in combination with an industrial manipulator, to test how AR can affect the workload of programmers, who teach the robot all necessary movements. In particular, Sphero 2.0 robot ball was used in combination with handheld AR in a form of a tablet to visualize and control Sphero's movement (Fig. 7). While the results show that the completion time of programmers' tasks is increased, the mental demand is decreased.

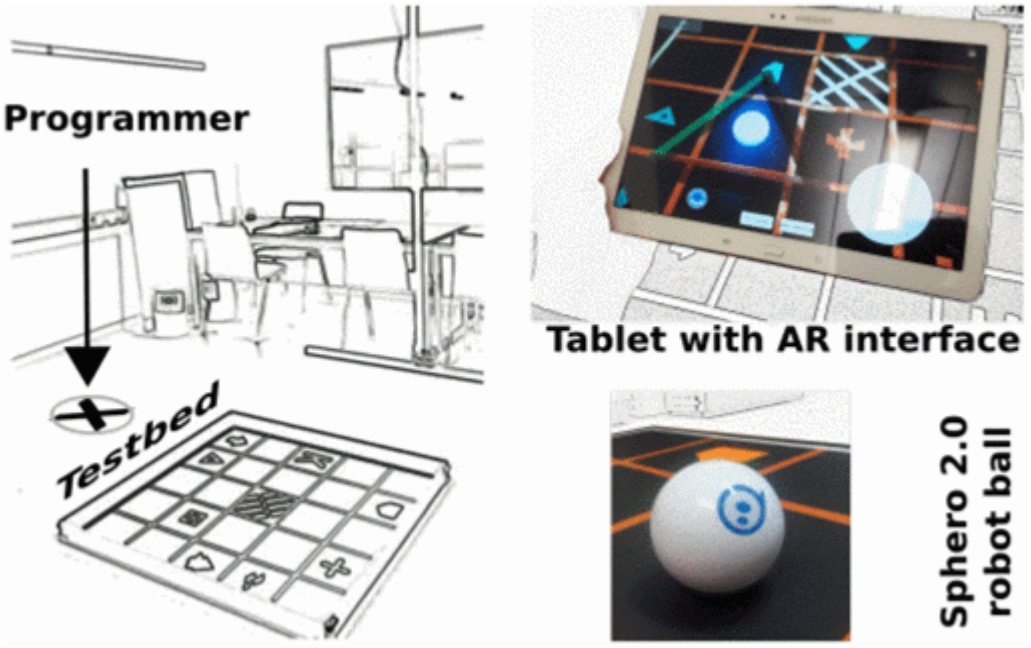


Figure 7: Setup, used test, how AR affects workload [23]

In [24] a combination of AR and Unmanned Aerial Vehicle (UAV) is used to enhance 3D scanning of environments. The paper proposes to use this solution in environments, where some prior knowledge exists. User can preview in AR the 3D environment, which the UAV is mapping. Based on the seen map user can help UAV by teleoperating it into areas not yet mapped, by tilting the head towards different directions (Fig 8).

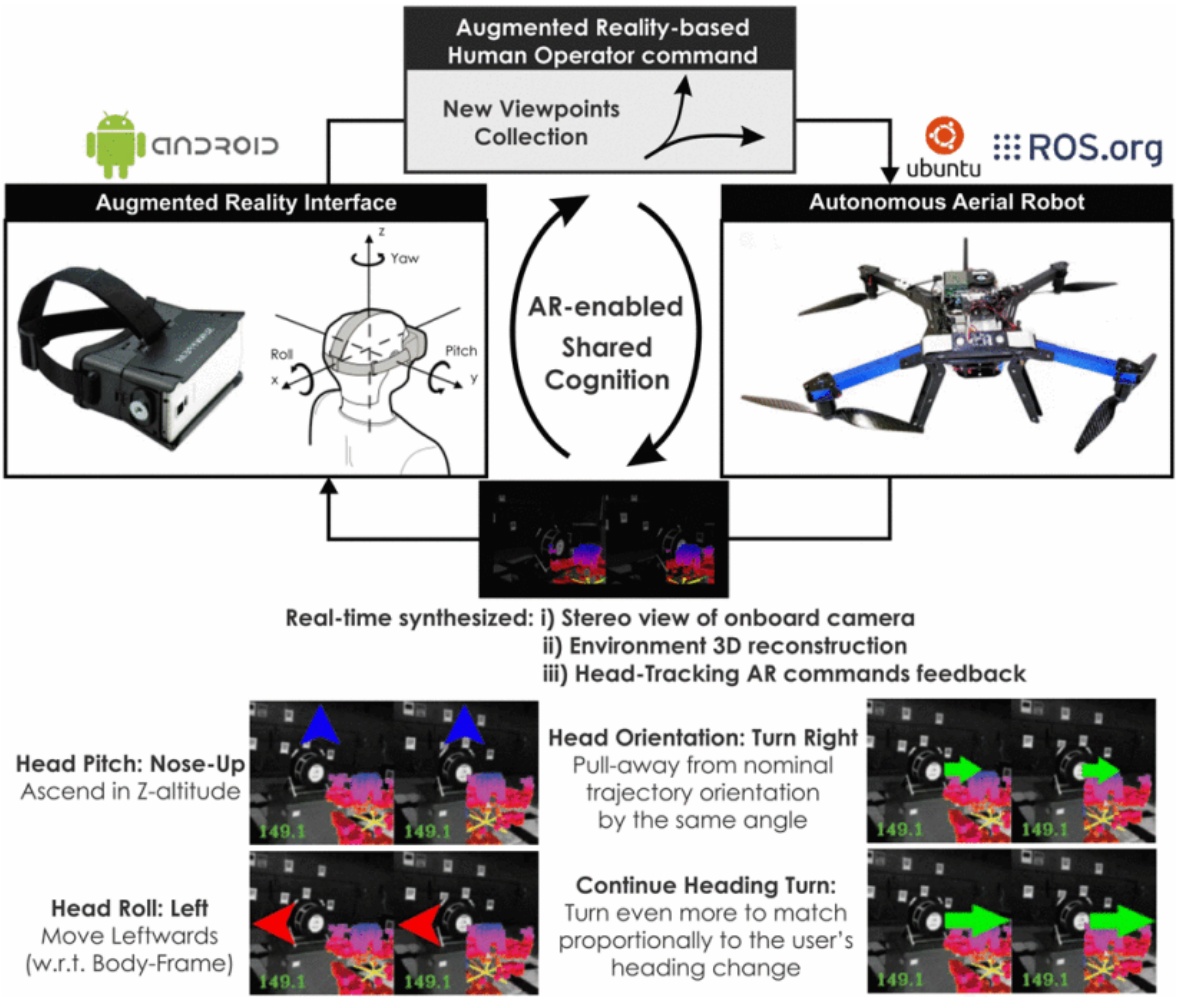
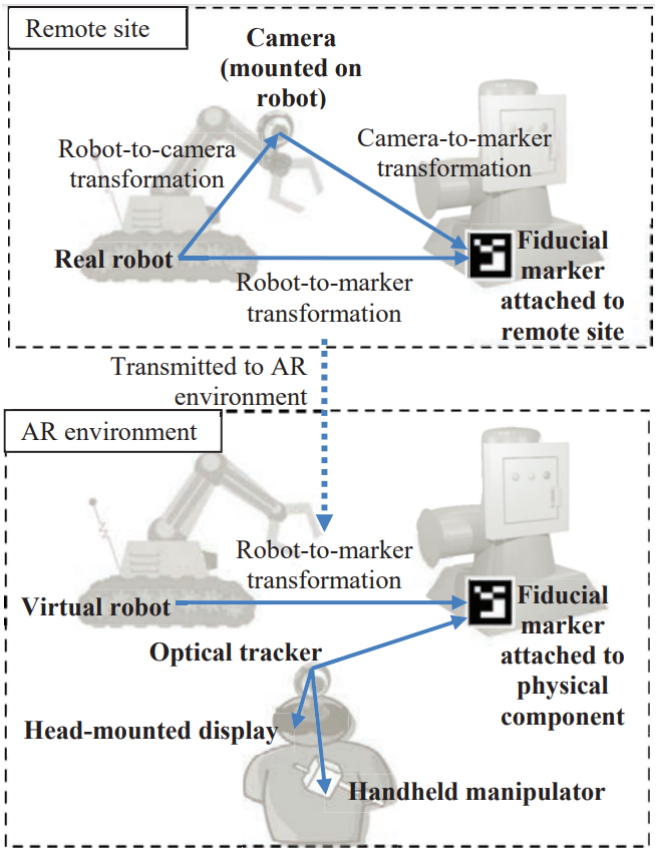
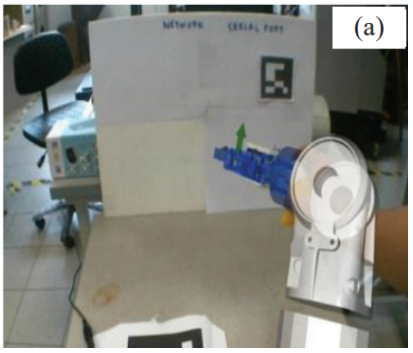


Figure 8: Overview of AR interface, used for mapping the environments [24]

For teleoperation, [25] proposes the use of AR to increase operators' spatial awareness by visualizing information, such as the robot state and the surrounding, visible by the camera, mounted on the robot in relation to corresponding components, thus reducing mental demand of the operator (Fig. 9).



(a)



(b)

Figure 9: Proposed by [25] architecture for teleoperation using AR. (a) with real-life use case; (b) robot visualization

One of the interesting solutions in the field is proposed by [26] in a form of an AR system to help operators perform maintenance on the robot in case of any physical failures. The proposed solution uses AR to display instructions on how to perform specific maintenance on the robot (Fig. 10).

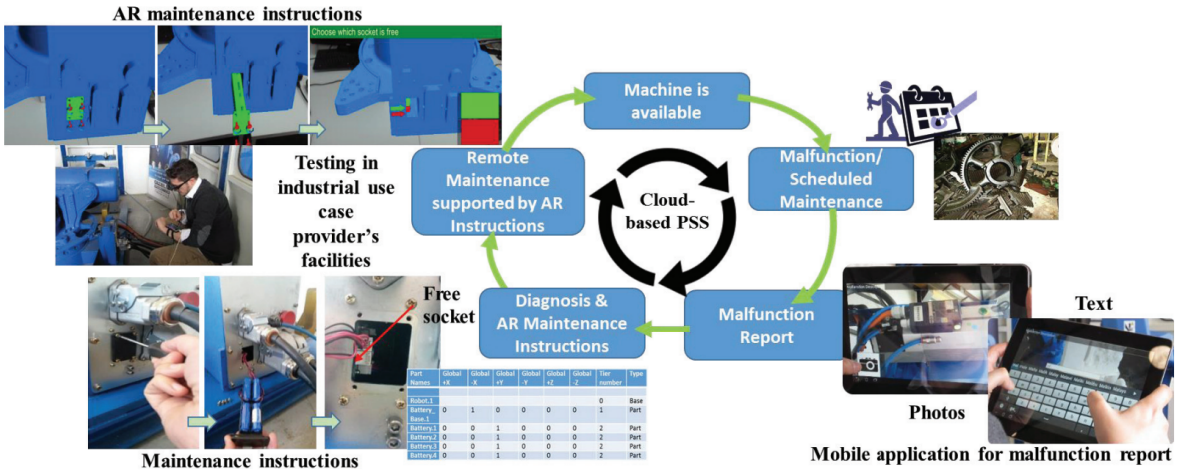


Figure 10: Proposed by [26] Product Service System with the use of AR

One of the main points of AR is ensuring the safety of Human-Robot Interaction (HRI) by having the operator work or interact with a virtual counterpart, rather than a real robot. One of the solutions, currently being developed uses Unity engine and ROS to visualize robot manipulator state and movement (Fig. 11) [27].

Figure 11: Demonstration of robot manipulator, used by Unity Technologies for ROS-Unity integration showcase [28] and also as demo manipulator in [27]

Projection-based AR can also be used for such purposes, as [17] uses projections, to display and warn operator about manipulator intentions (Fig. 12).

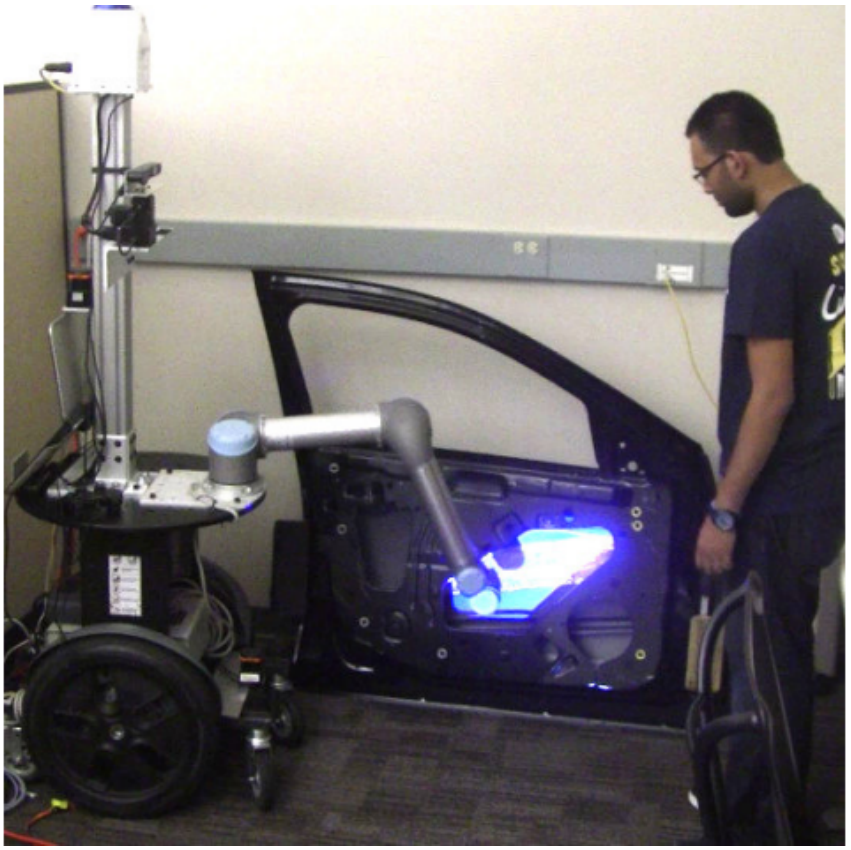
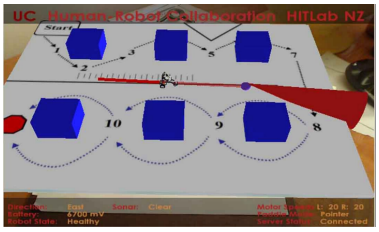
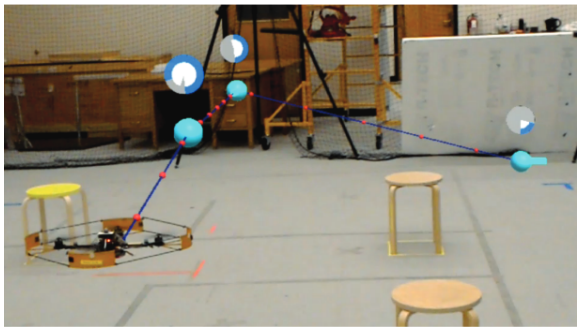


Figure 12: Using projections to show the area where the robot is going to do its work. [17]

There are also papers, such as [29] in the case of UAV (Fig. 13b), where AR is used to project the robot's future trajectory, thus letting operators know, in which areas they should be careful. In [18], [30] AR is used in combination with Mobile Robots (MR). In [30] AR shows robot movement in the virtual environment, thus relying on the remote control (Fig. 13a). AR in [18] is using projections on the floor to display robots' movement intentions and thus giving surrounding people the warning to be careful when moving in that direction (Fig. 13c).



(a)



(b)

(c)

Figure 13: Usage of AR for robot trajectory visualization. (a) AR is used to show the position and trajectory of the mobile robot [30]; (b) AR is used to display the trajectory and navigation goals of the drone [29]; (c) projecting robots movement orientation on the floor [18]

3.3 Tools for integrating AR and robotics

3.3.1 ROS

Robot Operating System (ROS) is an open-source data distribution software framework and a set of tools, which allows to make robotic applications and port them on any robot, which uses ROS in its backend [31].

ROS Navigation and MoveIt are two separate collections of software libraries and tools that assist in developing autonomous behavior for mobile robots and manipulators, respectively. ROS Navigation framework uses odometry and sensors, such as depth cameras, lidars and laserscans to localize the robot. ROS Navigation can receive navigation goals and, based on robot's localized position, generate velocity commands to reach it [32]. Moveit is a

manipulator motion planning framework, which includes motion planners, kinematic solvers, and other tools to deploy manipulators [33].

3.3.2 Unity

Unity is a game engine, being developed by Unity Technologies. Its monetization model allows anyone to use it for free if their company has less than 100,000 $ of yearly revenue [34]. Unity supports all major platforms, including PC in a form of Windows, Linux and Mac, mobile in a form of iOS and Android, game consoles and XR in a form of both VR and AR [35]. A combination of both of these aspects make Unity a popular choice for developers [36].

While Unity has support for the Universal Windows Platform (UWP) app format, developed by Microsoft to run apps on Windows 10 and 11 machines, including Hololens 2, it does not support AR development by default. For that purpose, Microsoft developed Mixed Reality Toolkit (MRTK), which allows Unity developers to write apps that will work with Microsoft's lineup of VR and AR devices, including the Hololens 2. MRTK provides all the necessary scripts and events to allow developers to process all input types, supported by Hololens, and write and deploy apps [37].

3.3.3 ROS-Unity communication

By default Unity and ROS cannot exchange data. There are 3 major solutions developed to overcome this issue: ROS#, MirrorLabs, and Unity-Robotics-Hub.

ROS# (ros-sharp)

The ros-sharp, developed by Siemens [38] uses the ros_bridge package from ROS and websocket in Unity to establish communication and allow ROS and Unity to exchange ROS messages. It has a set of default ROS message types built-in, but developers can import new and custom message types. It also allows importing ROS robot models described in the Universal Robot Description Format (URDF). The downside is that it does not with UWP apps, which is a requirement for developing for Hololens 2. Even though a git fork with UWP support has been created from the ros-sharp source code, the latest update outside the readme-file is from June 9th, 2021 [39] and does not work out-of-box with the current long-term Unity 2019 version, 2019.4.39f1. The original ros-sharp repository continues to be regularly updated [38].

MirrorLabs

MirrorLabs is an EIT Manufacturing activity that developed software and educational content for combining AR/VR with existing robotic infrastructure [40]. During the MirrorLabs project, the framework for ROS-Unity app development was created [41], [42]. It consists of a modified version of ros-sharp and a version of MRTK, which works on Unity 2019.4.18f1, used in this framework. This removes the necessity to install MRTK and ros-sharp manually. Same as in ros-sharp, it can import ROS robot models into Unity scene from a URDF description file. The downside is that there is no support for importing new ROS message types, which limits the communication to built-in message types.

Unity-Robotics-Hub

In recent years Unity Technologies have grown interested in ROS and, as a result, developed their own solution for ROS-Unity communication [28]. It supports UWP apps as well as importing URDF and custom message types [43]. Furthermore, there is support for ROS2 and the source code repository is being actively updated [44]. The current goal of Unity Technologies is to allow the Unity engine to work as a physical simulation, similar to the Gazebo simulator [45], which ROS uses by default. Implementation of all sensors and actuators is still a work in progress, meaning it is still cannot be fully used as a robotic simulator [44].

Table 1 shows the summary of the most important points about each solution. Whether it supports Hololens so that a user interface created with this solution will work on AR headsets. If any ROS message type can be imported, allowing the developer to use data from sensors, such as laserscan or topics, such as map, message types for which are not installed in any of them by default. The last one is whether or not the solution comes with a pre-configured setup of Unity, where both MRTK and the solution itself are installed.

Solution name	Support for Hololens	Import any ROS message to Unity	Preconfigured Unity setup
ROS#		✓	
MirrorLabs	✓		✓
Unity-Robotics-Hub	✓	✓	

Table 1: Checklist with the functionality of each ROS-Unity communication solution.

4. Technical requirements

The goal of the thesis is to construct an AR interface, which would allow the user to interact with robotic systems, such as manipulators or mobile robots (MR).

4.1. Functional requirements

1. A user can designate the goal location for a mobile robot relative to the physical location of the MR. The goal marker is visualized on the display of the AR headset.
2. The motion plan generated by the ROS move_base controller is visualized relative to the physical robot on the display of the AR headset.
3. A user can use their hand to designate the target state for a manipulator robot relative to the physical manipulator robot.
4. The motion plan generated by ROS MoveIt is visualized relative to the physical manipulator on the display of the AR headset.
5. The interface can also be used on simulated robots.

4.2. Non-functional requirements

6. Microsoft Hololens 2
7. ROS Noetic
8. Unity 2020 LTS
9. The system should be universal so that it can be deployed on robots from different manufacturers.

5. Design

During the thesis, 2 solutions were developed. The first one is for manipulators as part of the MirrorLabs project. It consists of 2 repositories: the first one contains the Unity project [46] and the second one contains the ROS package, required for the interface to work [47]. The second one is for control and interaction with mobile robots. It consists of the repository with the Unity project [48].

5.1 AR interface for commanding robot manipulators

The idea is to replicate the user experience of operating a manipulator through the MoveIt GUI. It means that, in AR, the user sees 2 manipulators of different colours. The first one is of the same colour, as a real manipulator and is responsible for visualizing a real robot joint state. If a real or simulated robot will move, this one will move as well. The second robot is of different colour, which is by default orange and also has a green virtual ball in the place of the robot's end-effector (eef). The user can grab the ball and move and rotate it in space, thus setting the eef position, to which the robot will need to go in the future. During the process of setting eef position, the orange manipulator will change its joint state to accommodate the desired eef pose, thus allowing the user to preview a future robot state. After the user is satisfied with the pose, the virtual ball can be released from the hand and the real robot will start moving to achieve the same joint state as the orange one, also meaning the same eef position. System design can be seen in Fig. 14.

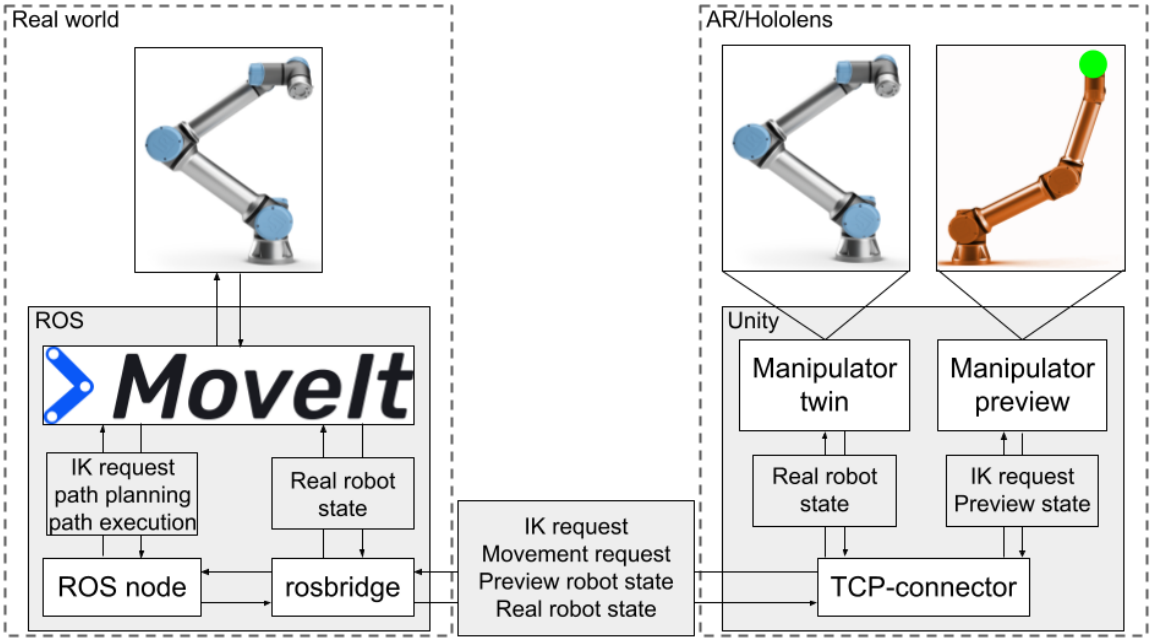


Figure 14. Design of the AR interface for manipulator control.

The interface was originally created as a test of what can be done, using the framework, developed in MirrorLabs. Manipulator, used for this application is UR5 from Universal Robots.

The unity-based user interface consists of 4 main components:

1) operator, who is interacting with the scene,
2) manipulator model visualizing the position of the real robot,
3) second manipulator model with intractable eef, which previews the goal state of the manipulator,
4) and websocket, which manages communication between ROS and Unity.

The system is designed so that the operator is able to move the whole scene around and interact with an object, representing eef position in cartesian space. These eef positions are sent to the ROS side through websocket. ROS side computes the inverse kinematics and sends corresponding joint states to Unity to update the preview manipulator. A pose is deemed acceptable after the operator releases the eef virtual marker. Then ROS side uses MoveIt MoveGroup C++ Interface to generate a motion plan between the current and preview states as well as executes the movements of the real robot (Fig. 14).

ROS side consists of 2 parts: MoveIt and ROS command node. Command node receives the eef position from Unity/Hololens and computes Inverse Kinematics (IK) of

preview manipulator and sends it back on the corresponding topic. After information about stopping interaction with the ball is received, the node requests MoveIt to compute the trajectory for the real manipulator to move to a new state and after that requests to execute it (Fig. 15). MoveIt is responsible for publishing the real robot joint state on the corresponding topic, communicating with the real robot and executing requests from the ROS command node.

```
eef_position new_position(new_message) #called every time new position is sent by Unity
{
   return new_message.position
}
while (roscore running)
{
   joint_state = moveit_compute_IK(eef_position)
   publish(joint_state)
   if (new_message.user_satisfied == 1)
   {
      plan = make_plan_to(joint_state)
      execute(plan)
   }
}
```

Figure 15: Working principle of ROS manipulator command node

5.2 AR interface for commanding mobile robots

Goal of the interface is to provide the user with the ability to send navigation goals, which are relative to the real robot's position and display additional information, such as robot's movement trajectory. Fig. 16 depicts the idea of how the interface is intended to be used.

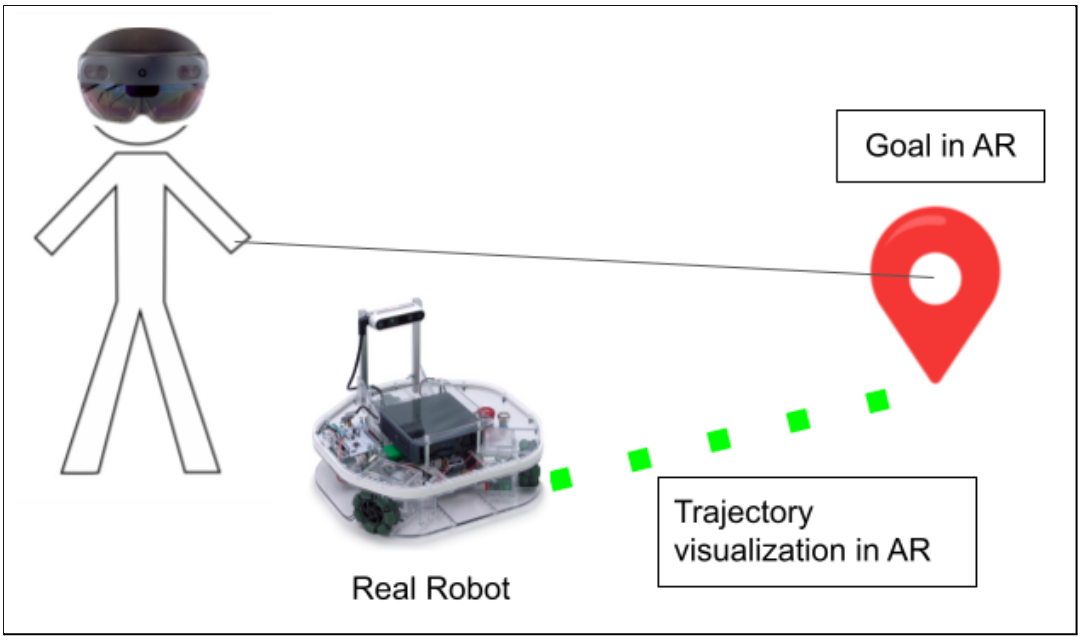


Figure 16: Graphics depicting the user experience of AR interface for commanding MR. The operator can grab the corresponding object to set a goal for the real robot and see the real robot's planned trajectory visualized in AR.

For this interface, Unity-Robotics-Hub was chosen for communication between ROS and Unity. The reason is that the MirrorLabs framework concentrates on the manipulators and does not support the importing of new messages, such as trajectory messages or map. On the other hand, Unity's solution works also with custom ROS messages and supports UWP out of the box, which makes it the only working option at the moment (Table 1).

The ROS part of the application consists of the ROS Navigation, TCP endpoint, which is part of the Unity-Robotics-Hub and is used to connect ROS to Unity, and static transform publisher between the robot's map and real-life QR code.

QR code is needed to correctly place all visualized information in space. Because of the Navigation toolkit in ROS, the robot knows its state in the mapped environment. The idea is to place the QR code in the predefined location in real life and specify the corresponding pose in the ROS coordinate frame transformation tree, which describes the poses of all objects relative to each other.

Hololens can read the QR code in a real environment and by using the transformation tree, place all virtual objects in the correct positions.

The Unity side includes an operator, interacting with the scene, QR code reader, which parses the location of QR code and the stored name of ROS transformation frame, robot, displayed according to odom topic of ROS and having on the option to be turned off, local

and global trajectories, which are displayed during robots autonomous movement, map and navigation goal, which operator sends to robots navigation.

Fig. 17 depicts the working principle of the AR interface for MR, including both ROS and Unity sides of things. Communication is done by TCP connectors, both sides of which are created by Unity Technologies [43].

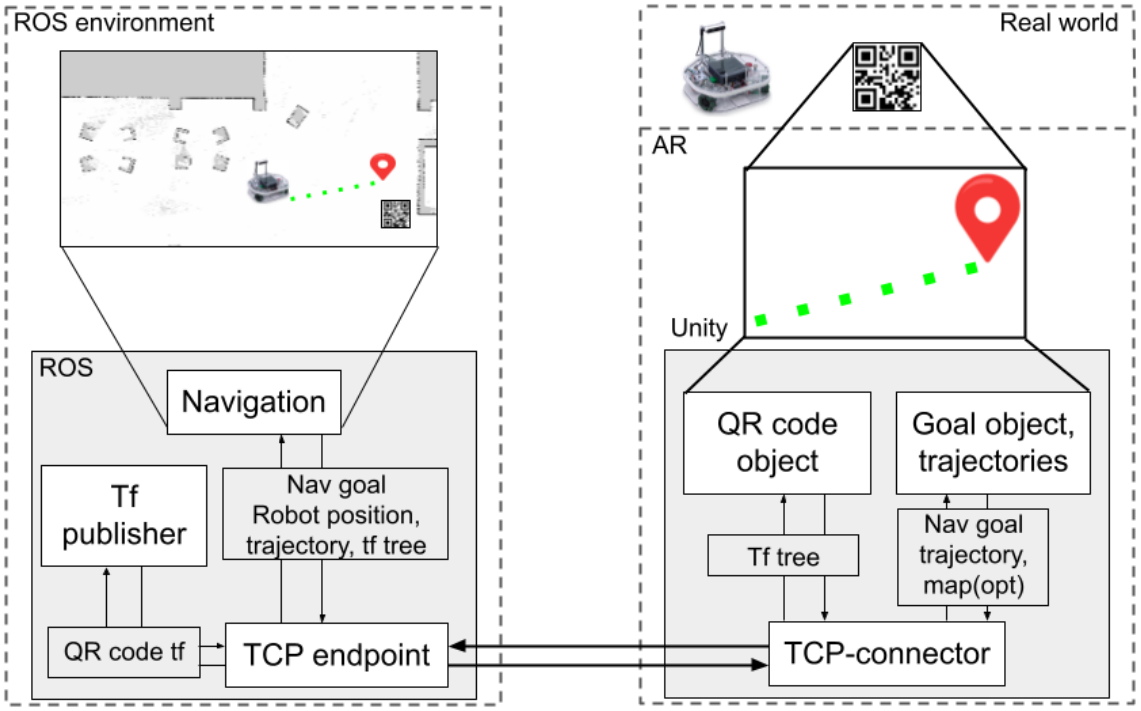


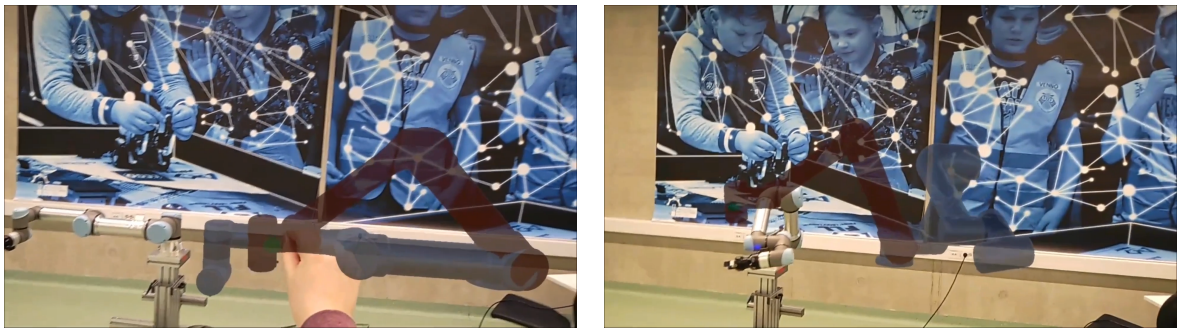
Figure 17: AR interface schematic of both ROS and Unity sides.

By relying on the ROS Navigation for motion planning, the robot which is used and visualized can be easily changed, by importing a new URDF model in Unity.

6. Results

6.1 Manipulator application

To run the interface on Hololens, the user must build it into the UWP app and deploy it onto Hololens. Before building the Unity project, the user will define the IP of the ROS machine, which controls the manipulator. After deploying it, and making sure that both the Hololens and the ROS computer are in the same network, the operator can launch the MoveIt interface on the computer and start the deployed application on Hololens. During the interaction, orange UR5 will preview the future robot manipulator position, the same way it is done in MoveIt. Upon releasing the eef, the robot will start moving to the position, that the user made (Fig. 19).



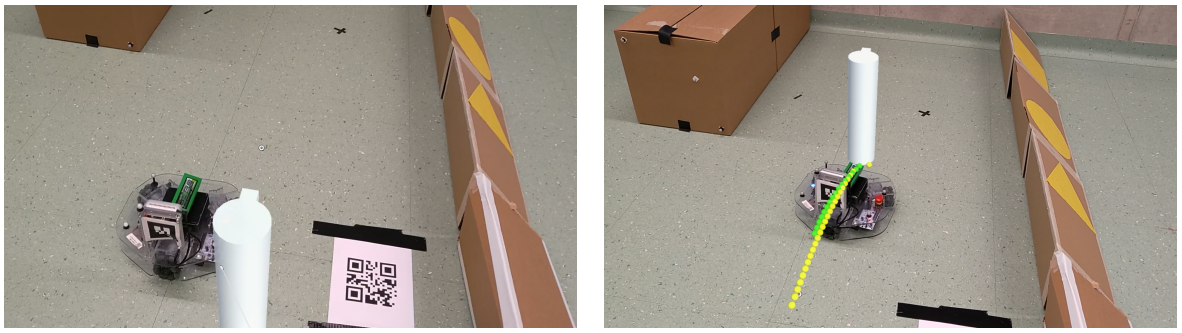
(a) (b)

Figure 19: Interaction of operator with virtual UR5 (a) while defining the goal with arm preview and (b) during robot movement to newly defined location

6.2 Mobile robot application

To run the interface on Hololens, the user must build it into the UWP app and deploy it onto Hololens. Before building the project, the operator can change which robot model is going to be used. It is done by importing a new URDF model and assigning it to the script, responsible for odom data. If the interface is planned to be used with a real robot, model visualization can be turned off. The operator can also decide, if map visualization is necessary or not, depending on whether network bandwidth is fast enough.

Upon starting the deployed application and setting up the robot or simulation, the operator can launch the application from Hololens. After launching the application and finding the QR code, it will be possible to send a navigation goal. After sending such a goal, the robot will start moving toward it and Hololens will display the robot's global trajectory in yellow dots and local trajectory in green (Fig. 20).



(a) (b)

Figure 20: Unity application to control MR (a) before sending the command with goal object and QR code visible and (b) during movement with global trajectory (yellow) and local trajectory (green) visualized

7. Summary

Outcome of the thesis is 2 Unity AR applications for HRI. The first application is constructed to send the end-effector goal to the manipulator and preview its position.

The second application is designed to control any mobile robot, which uses ROS Navigation. The AR interface of this application can be used in co-location with the real robot in the same environment and in stand-alone for remote control of the robot.

8. Acknowledgements

I would like to express my gratitude to my supervisor, Karl Kruusamäe, Associate professor at the Institute of Technology, Tartu University, for helping and encouraging me during both the theoretical and practical parts of my thesis and for giving me valuable advises. I would also like to thank Veiko Vunder for his help with understanding the concepts of 3D graphics. And want to express my gratitude to my family and friends for encouraging and believing in me during my time as a master's student.

Rybalskii Igor 

9. Bibliography

https://github.com/ut-ims-robotics/unity_ros_navigation_demo

[1] H. Lasi, P. Fettke, H.-G. Kemper, T. Feld, and M. Hoffmann, “Industry 4.0,” *Bus. Inf. Syst. Eng.*, vol. 6, no. 4, pp. 239–242, Aug. 2014, doi: 10.1007/s12599-014-0334-4.

[2] X. Xu, Y. Lu, B. Vogel-Heuser, and L. Wang, “Industry 4.0 and Industry 5.0—Inception, conception and perception,” *J. Manuf. Syst.*, vol. 61, pp. 530–535, Oct. 2021, doi: 10.1016/j.jmsy.2021.10.006.

[3] “Standards - IEEE Digital Reality.” https://digitalreality.ieee.org/standards (accessed Mar. 28, 2022).

[4] Z. Makhataeva and H. A. Varol, “Augmented Reality for Robotics: A Review,” *Robotics*, vol. 9, no. 2, Art. no. 2, Jun. 2020, doi: 10.3390/robotics9020021.

[5] “HoloLens 2—Overview, Features, and Specs | Microsoft HoloLens.” https://www.microsoft.com/en-us/hololens/hardware (accessed Mar. 28, 2022).

[6] “Magic Leap 1 AR information and pricing | Magic Leap.” https://www.magicleap.com/en-us/magic-leap-1 (accessed Mar. 28, 2022).

[7] “About Raptor – Everysight.” https://everysight.com/about-raptor/ (accessed Mar. 28, 2022).

[8] K. Melnick, “AR Functionality Comes To Vive Pro With New Front-Facing Camera Tools,” *VRScout*, Apr. 17, 2018. https://vrscout.com/news/ar-functionality-vive-pro-front-facing-camera-tools/ (accessed Apr. 18, 2022).

[9] UploadVR, *HoloLens 2 AR Headset: On Stage Live Demonstration*, (2019). Accessed: Apr. 18, 2022. [Online Video]. Available: https://www.youtube.com/watch?v=uIHPPtPBgHk

[10] “Pokémon Go launches in U.S. on iOS and Android,” *VentureBeat*, Jul. 07, 2016. https://venturebeat.com/2016/07/06/pokemon-go-launches-worldwide-on-ios-and-android/ (accessed Mar. 28, 2022).

[11] “Pokémon Go Revenue and Usage Statistics (2022),” *Business of Apps*, Aug. 08, 2017. https://www.businessofapps.com/data/pokemon-go-statistics/ (accessed Mar. 28, 2022).

[12] K. Wagner, “What is Pokémon Go and why is everybody talking about it?,” *Vox*, Jul. 09, 2016. https://www.vox.com/2016/7/9/12132748/what-is-pokemon-go-game (accessed Apr. 18, 2022).

[13] “AR ruler - Android Apps on Google Play.” https://play.google.com/store/search?q=AR%20ruler&c=apps&hl=en&gl=US (accessed Mar. 28, 2022).

[14] “Ikea App Page.” https://www.ikea.com/au/en/customer-service/mobile-apps/say-hej-to-ikea-place-pub1f8af050 (accessed Mar. 28, 2022).

[15] “Download the Wayfair App Today!,” *Wayfair*. https://www.wayfair.com/the-wayfair-app (accessed Mar. 28, 2022).

[16] A. Doshi, R. T. Smith, B. H. Thomas, and C. Bouras, “Use of projector based augmented reality to improve manual spot-welding precision and accuracy for automotive manufacturing,” *Int. J. Adv. Manuf. Technol.*, vol. 89, no. 5–8, pp. 1279–1293, Mar. 2017, doi: 10.1007/s00170-016-9164-5.

[17] R. S. Andersen, O. Madsen, T. B. Moeslund, and H. B. Amor, “Projecting robot intentions into human environments,” in *2016 25th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, Aug. 2016, pp. 294–301. doi: 10.1109/ROMAN.2016.7745145.

[18] M. D. Coovert, T. Lee, I. Shindev, and Y. Sun, “Spatial augmented reality as a method

for a mobile robot to communicate intended movement," *Comput. Hum. Behav.*, vol. 34, pp. 241–248, May 2014, doi: 10.1016/j.chb.2014.02.001.
[19] "Tech Specs," *Glass*. https://www.google.com/glass/tech-specs/ (accessed May 16, 2022).
[20] "Case Studies," *Glass*. https://www.google.com/glass/case-studies/ (accessed May 16, 2022).
[21] "V11H756020 | Moverio BT-300 Smart Glasses (AR/Developer Edition) | Smart Glasses | Wearables | For Work | Epson US." https://epson.com/For-Work/Wearables/Smart-Glasses/Moverio-BT-300-Smart-Glasses-%28AR-Developer-Edition%29-/p/V11H756020 (accessed May 16, 2022).
[22] "Glass," *Glass*. https://www.google.com/glass/start/ (accessed May 16, 2022).
[23] S. Stadler, K. Kain, M. Giuliani, N. Mirnig, G. Stollnberger, and M. Tscheligi, "Augmented reality for industrial robot programmers: Workload analysis for task-based, augmented reality-supported robot control," in *2016 25th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, Aug. 2016, pp. 179–184. doi: 10.1109/ROMAN.2016.7745108.
[24] C. Papachristos and K. Alexis, "Augmented reality-enhanced structural inspection using aerial robots," in *2016 IEEE International Symposium on Intelligent Control (ISIC)*, Sep. 2016, pp. 1–6. doi: 10.1109/ISIC.2016.7579983.
[25] A. W. W. Yew, S. K. Ong, and A. Y. C. Nee, "Immersive Augmented Reality Environment for the Teleoperation of Maintenance Robots," *Procedia CIRP*, vol. 61, pp. 305–310, 2017, doi: 10.1016/j.procir.2016.11.183.
[26] D. Mourtzis, V. Zogopoulos, and E. Vlachou, "Augmented Reality Application to Support Remote Maintenance as a Service in the Robotics Industry," *Procedia CIRP*, vol. 63, pp. 46–51, Jan. 2017, doi: 10.1016/j.procir.2017.03.154.
[27] C. Xue, Y. Qiao, and N. Murray, "Enabling Human-Robot-Interaction for Remote Robotic Operation via Augmented Reality," in *2020 IEEE 21st International Symposium on "A World of Wireless, Mobile and Multimedia Networks" (WoWMoM)*, Aug. 2020, pp. 194–196. doi: 10.1109/WoWMoM49955.2020.00046.
[28] "Unity-Robotics-Hub/README.md at main · Unity-Technologies/Unity-Robotics-Hub," *GitHub*. https://github.com/Unity-Technologies/Unity-Robotics-Hub (accessed Apr. 25, 2022).
[29] M. Walker, H. Hedayati, J. Lee, and D. Szafir, "Communicating Robot Motion Intent with Augmented Reality," in *Proceedings of the 2018 ACM/IEEE International Conference on Human-Robot Interaction*, Chicago IL USA, Feb. 2018, pp. 316–324. doi: 10.1145/3171221.3171253.
[30] S. A. Green, X. Q. Chen, M. Billinghurst, and J. G. Chase, "Collaborating with a Mobile Robot: An Augmented Reality Multimodal Interface," *IFAC Proc. Vol.*, vol. 41, no. 2, pp. 15595–15600, 2008, doi: 10.3182/20080706-5-KR-1001.02637.
[31] M. Quigley *et al.*, "ROS: an open-source Robot Operating System," p. 6.
[32] "navigation - ROS Wiki." http://wiki.ros.org/navigation (accessed May 16, 2022).
[33] D. Coleman, I. S. Ucan, S. Chitta, and N. Correll, "Reducing the Barrier to Entry of Complex Robotic Software: a MoveIt! Case Study," p. 14, 2014.
[34] G. Cuofano, "How Does Unity Work And Make Money? Unity Business Model Explained," *FourWeekMBA*, Oct. 11, 2020. http://fourweekmba.com/unity-business-model/ (accessed May 02, 2022).
[35] "What platforms are supported by Unity?," *Unity*. https://support.unity.com/hc/en-us/articles/206336795-What-platforms-are-supported-by-Unity- (accessed May 02, 2022).
[36] "Technologies," *SteamDB*. https://steamdb.info/tech/ (accessed May 02, 2022).

[37] *What is the Mixed Reality Toolkit*. Microsoft, 2022. Accessed: May 16, 2022. [Online]. Available: https://github.com/microsoft/MixedRealityToolkit-Unity
[38] *siemens/ros-sharp*. Siemens, 2022. Accessed: May 02, 2022. [Online]. Available: https://github.com/siemens/ros-sharp
[39] E. Voll, *EricVoll/ros-sharp*. 2022. Accessed: May 02, 2022. [Online]. Available: https://github.com/EricVoll/ros-sharp
[40] "MirrorLabs – creating similar learning environment for students all over Europe for human-robot coproduction," *EIT Manufacturing*, Mar. 18, 2020. https://www.eitmanufacturing.eu/news-media/activities/mirrorlabs-creating-similar-learning-environment-for-students-all-over-europe-for-human-robot-coproduction/ (accessed May 15, 2022).
[41] D. Aschenbrenner and J. Rieder, *Framework for the publication: MirrorLabs – creating similar learning environments for students all over Europe for human-robot coproduction*. 2021. [Online]. Available: https://data.4tu.nl/articles/code/Framework_for_the_publication_MirrorLabs_creating_similar_learning_environments_for_students_all_over_Europe_for_human-robot_coproduction/14186807/1
[42] D. Aschenbrenner *et al.*, "Mirrorlabs - creating accessible Digital Twins of robotic production environment with Mixed Reality," in *2020 IEEE International Conference on Artificial Intelligence and Virtual Reality (AIVR)*, Dec. 2020, pp. 43–48. doi: 10.1109/AIVR50618.2020.00017.
[43] *ROS TCP Connector*. Unity Technologies, 2022. Accessed: May 17, 2022. [Online]. Available: https://github.com/Unity-Technologies/ROS-TCP-Connector
[44] *Unity Robotics Hub*. Unity Technologies, 2022. Accessed: May 02, 2022. [Online]. Available: https://github.com/Unity-Technologies/Unity-Robotics-Hub
[45] *Ignition Gazebo : A Robotic Simulator*. Gazebo, 2022. Accessed: May 19, 2022. [Online]. Available: https://github.com/gazebosim/gz-sim
[46] *MirrorLabs_HL2*. Robotics at IMS Lab, 2021. Accessed: May 19, 2022. [Online]. Available: https://github.com/ut-ims-robotics/MirrorLabs_HL2
[47] *hl_ur5_ik*. Robotics at IMS Lab, 2021. Accessed: May 19, 2022. [Online]. Available: https://github.com/ut-ims-robotics/hl_ur5_ik
[48] *unity_ROS_navigation_demo*. Robotics at IMS Lab, 2021. Accessed: May 20, 2022. [Online]. Available: https://github.com/ut-ims-robotics/unity_ros_navigation_demo

10. Appendices

Supplementary archive with unity project

Description

The accompanying archive contains Unity Project with AR interface for manipulators.

Filename

MirrorLabs_HL2-main.zip

Supplementary archive with catkin package

Description

The accompanying archive contains a catkin package, meant to be used on ROS machine in combination with AR interface for manipulators

Filename

hl_ur5_ik-main.zip

Supplementary archive with unity project

Description

The accompanying archive contains Unity Project with AR interface for MRs.

Filename

unity_ros_navigation_demo-igor-devel.zip

Non-exclusive licence to reproduce the thesis and make the thesis public

I, Igor Rybalskii
(*author's name*)

1. grant the University of Tartu a free permit (non-exclusive licence) toreproduce, for the purpose of preservation, including for adding to the DSpace digital archives until the expiry of the term of copyright, my thesis

Augmented reality (AR) for enabling human-robot collaboration with ROS robots,
(*title of thesis*)

supervised by
Karl Kruusamäe.
(supervisor's name)

2. I grant the University of Tartu a permit to make the thesis specified in point 1 available to the public via the web environment of the University of Tartu, including via the DSpace digital archives, under the Creative Commons licence CC BY NC ND 4.0, which allows, by giving appropriate credit to the author, to reproduce, distribute the work and communicate it to the public, and prohibits the creation of derivative works and any commercial use of the work until the expiry of the term of copyright.

3. I am aware of the fact that the author retains the rights specified in points 1 and 2.

4. I confirm that granting the non-exclusive licence does not infringe other persons' intellectual property rights or rights arising from the personal data protection legislation.

Igor Rybalskii
24/05/2022