This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 820767

CoLLaboratE

Co-production CeLL performing Human-Robot Collaborative AssEmbly

D4.5 – Mobile interface to teach robot paths with variations (preliminary)

Due date: M12

Abstract:

This deliverable presents a preliminary version of the software for a mobile interface to teach robot paths with variations. Preliminary software functionalities include the selection of points on the screen of the smartphone, the visualization of the trajectories taught by kinesthetic teaching of the robot and the visualization of the robot. Implementations on a Huawei Mate20 Pro smartphone and Panda robot show how it can already be exploited in CoLLaboratE use-cases and within a wide range of industrial tasks.

Dissemination Level		
PU	Public	x
PP	Restricted to other programme participants (including the Commission Services)	
RE	Restricted to a group specified by the consortium (including the Commission Services)	
CO	Confidential, only for members of the consortium (including the Commission Services)	

Document Status

Document Title	D4.5 – Mobile interface to teach robot paths with variations (preliminary)
Version	1.0
Work Package	4
Deliverable	4.5
Prepared by	IDIAP
Contributors	KOLEKTOR
Checked by	BOR, LMS
Approved by	AUTH
Due Date	30/09/2019
Completion Date	30/09/2019
Confidentiality	Public

Document Change Log

Each change or set of changes made to this document will result in an increment to the version number of the document. This changes log records the process and identifies, for each version number of the document, the modification(s) which caused the version number to be incremented.

Change Log	**Version**	**Date**
First draft	0.1	September 27, 2019
Contributing partners' input	0.2	September 30, 2019
Internal reviewers' feedback	0.3	September 30, 2019
Final	1.0	September 30, 2019
Fixed amended information and re-uploaded	1.1	November 1, 2019

EXECUTIVE SUMMARY

The present document is a deliverable of the CoLLaboratE project, funded by the European Commission's Directorate-General for Research and Innovation (DG RTD), under its Horizon 2020 Research and innovation program (H2020). This deliverable will present a preliminary version of the software for a mobile interface to teach robot paths with variations. Augmented Reality (AR) technologies, specifically open-source ARCore software will be used to develop an Android application to be exploited in CoLLaboratE. It includes functionalities such as selection of points by clicking on the screen of the smartphone and visualization of the trajectories learned by demonstration and the visualization of the robot.

The developed software has various software and hardware requirements to be applied on a real Human-Robot Collaboration case. The software can only be run on recent smartphones that are compatible with ARCore. It can be utilized for any tasks, with any robot communicating through Robot Operating System (ROS) nodes.

In this preliminary version, the user can run the application on the smartphone to (*i*) select points on the screen to calibrate with the robot, (*ii*) visualize learned robot trajectories and (*iii*) visualize the robot. Eliminating the need for a simulation environment, this application will help the users to (re)program the robots very easily and survey their movements by understanding better their intentions when adapting to different conditions, by executing them first on the screen of the smartphone. Future work will focus on exploiting learning from demonstration methods, specifically using the models provided by T3.3: Extension of movement primitives to behavior primitives task in the WP3. We will investigate methods for teaching movements with variations encoded in these behavior primitives and CoLLaboratE use-cases in which this technology can be showcased.

Table of Contents

Executive Summary 3

List of Figures 5

Abbreviations and Acronyms 5

1 Introduction 6

2 Requirements 6

2.1 Hardware 6

2.2 Software 7

3 Methods 8

3.1 Selection of points 8

3.2 Visualization of trajectories 8

3.3 Visualization of the robot 9

4 Discussions and future work 10

References 11

LIST OF FIGURES

Figure 1. Prototype of the developed smartphone interface to let an operator move the robot TCP to a desired location in an intuitive manner 6

Figure 2. Selection of a point 8

Figure 3. Visualization of a robot trajectory 9

Figure 4. Visualization of the robot 10

ABBREVIATIONS AND ACRONYMS

Partner's short name	Partner's full name
AUTH	ARISTOTLE UNIVERSITY OF THESSALONIKI
CERTH	CENTRE OF RESEARCH AND TECHNOLOGY HELLAS
ARMINES	ASSOCIATION POUR LA RECHERCHE ET LE DEVELOPMENT DES METHODES ET PROCESSUS INDUSTRIELS
JSI	INSTITUT JOZEF STEFAN
IDIAP	FONDATION DE L'INSTITUT DE RECHERCHE
UNIGE	UNIVERSITA DEGLI STUDI DI GENOVA
KU Leuven	KATHOLIEKE UNIVERSITEIT LEUVEN
LMS	UNIVERSITY OF PATRAS
CRF	CENTRO RICERCHE FIAT SOCIETA CONSORILE PER AZIONI
BOR	BLUE OCEAN ROBOTICS
ASTI	AUTOMATISMOS Y SISTEMAS DE TRANSPORTE INTERNO SA
KOL	KOLEKTOR ORODJARNA NACRTOVANJE IN IZDELAVA ORODIJ TER ORODJARSKE STORITVE D.O.O.S
ARCELIK	ARCELIK A.S.
ROMAERO	ROMAERO S.A.

Abbreviation	Definition
WP	Work Package
D	Deliverable
EC	European Commission
EU	European Union

1 INTRODUCTION

Human-Robot Collaboration (HRC) aims to provide an ergonomic, safe and easy-to-work environment for the humans working with the robots. However, collaborative robots are highly complex machines, and developing a Human-Robot Interaction is therefore a challenge. This software's objective is to create an easy-to-use and intuitive interface that enables effective communication between the robot and the user, to facilitate the development of Human-Robot Interactions as in Figure 1.

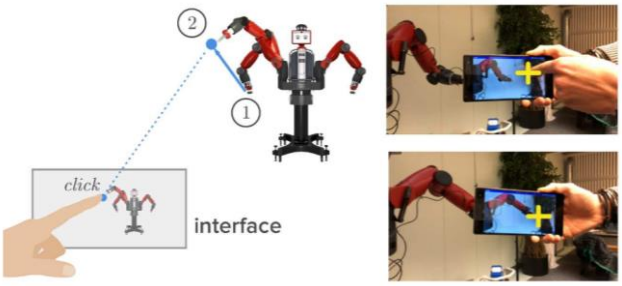

Figure 1. Prototype of the developed smartphone interface to let an operator move the robot TCP to a desired location in an intuitive manner

Augmented Reality (AR) is an interesting approach for creating this kind of interfaces. It consists of enhancing the real world by superimposing virtual information or objects over it. This research field is in constant evolution due to its numerous possibilities and applications, particularly in video games, education, medicine or industry, among others. Indeed, enhancing our perception of the world may allow us to ease some tasks or make them more interesting, leading to possible gains in precision and/or in productivity. AR technologies started to be applied increasingly in Industry 4.0 applications to help the workers to program the robots easily and to understand robot intentions more clearly [1] [2].

The purpose of the task "T4.3: Mobile interface to teach robot paths with variations" is to explore the use of AR capability on smartphones within HRC in industry, using Google ARCore platform. It enables a direct interaction between a smartphone and a robot, by visualizing its trajectories in the application, allowing a 3D representation of a given trajectory in the environment. Visualization of the robot itself helps the user to work with the robot even from far away. The final objective is to develop a mobile application that is as intuitive as possible, so that any non-expert user can control the robot, teach movements to perform tasks, visualize trajectories learned with the virtual robot performing them, before even executing on the real robot.

2 REQUIREMENTS

2.1 HARDWARE

This preliminary software is developed to work on any smartphone listed in [3], which is compatible with ARCore, an augmented reality software produced by Google [4]. The smartphone selected for this task is the HUAWEI Mate20 Pro [5], launched on the market in October 2018. It has a triple camera and a flash at the back, and a 3D depth sensing camera at the front, that enables

face recognition and 3D object scanning. The rear camera offers the widest focal length of all the recent smartphones' cameras, with a 35mm-equivalent focal length range from 16 to 80mm. The screen has a dimension of 6.4 inch, and a resolution of 1440 x 3120 pixel. ARCore exploits only a subset of these functionalities, where only a single camera is used at this stage of the development.

The robot used for the testing of the software is the FRANKA EMIKA Panda robot [6]. It is a seven-axis robotic arm which has joint position/velocity controllers embedded. It has torque sensors on each of its seven joints, and it is also back-drivable and gravity compensated, which facilitates HRC in an industrial context. It should be noted that the software developed can be used with any robot that is running Robot Operating System (ROS), by providing the robot parameters in a Unified Robot Description Format (URDF), file format.

2.2 SOFTWARE

In robotic applications, we need most of the time the robot to reach to a target pose, i.e. to perform a point-to-point motion. These motions can also be described as paths to follow, which requires the visualization of the trajectories. Finally, the visualization of the robot itself is required to check potential failures and for an effective communication between the user and the robot. ARCore software will be used for these purposes. ARCore is a Software Development Kit (SDK) developed by Google, which allows to build augmented reality applications on smartphones. Although ARCore can be used on all smartphones listed in [3], the preliminary software developed until M12 can only be used on the ones running Android operating system. ARCore is an open-source and free software that was released on 1st of March 2018. ARCore uses three main functions to integrate virtual content into the real world, via the phone's camera:

- Motion tracking allows the phone to estimate its position relatively to the world.
- Environmental understanding allows the phone to detect the size and the location of horizontal, vertical and inclined surfaces such as the ground, a table, or walls.
- Light estimation allows the phone to estimate the environment's current lighting conditions and the location of light sources.

When analyzing the environment, ARCore detects salient visual features and tracks their position over time. ARCore uses Concurrent Odometry and Mapping (COM), a method to continuously estimate the pose (position and orientation) of the smartphone with respect to the fixed world coordinate system. Two sources of information are combined for the highest accuracy possible of the estimation: the visual information given by the camera, which evaluates the displacement of the tracked features over time, and the inertial information given by the phone's Inertial Measurement Unit (IMU). If several features are detected on a same horizontal or vertical surface, ARCore recognizes this surface as a plane. It is important to notice that, as ARCore uses features to detect planes, a flat surface without texture, for example a white wall, may not be detected properly.

Visualization of the robot requires ROS, a widely used open-source robotics middleware that enables effective communication of the controller with the robot. ROS should either be installed in the robot, or in the controller.

3 METHODS

3.1 SELECTION OF POINTS

One of the goals of the interface is to command the robot to reach a specific position in the environment. In most of the robotics applications, we need the robot to perform point-to-point motions. This interface functionality solves the problem of the need to reprogram the robot to reach to a target position. It can be used with any industrial task containing this kind of movements, for example to command AGV's to reach to different destination. Selection of points serves also as a basis for the calibration of the robot with the environment which will be described briefly in Section 3.2.

We developed an interface to select a desired point by clicking on the screen of the smartphone, from at least two different views. The reason for choosing from at least two different views lies in the difficulty of inferring the depth information from 2D camera. The desired point in the environment is found by determining the closest intersection point of the virtual lines drawn by the user. If the lines are not intersecting and are not parallel, it will mark the midway of the closest distance between the lines. Since when the lines are parallel, it will find an intersection at the infinity, the user is required to draw the lines from different perspectives as shown in Figure 2. Figure 2 illustrates how the selection works using the application running on the smartphone. Black spot on the white paper represents the desired point to be selected. Clicking on the screen aiming for the black dot creates a virtual line passing through the dot. At least two non-parallel lines are required to define a virtual point on the screen, which requires moving the phone, so that two different viewpoints are used for the selection. A yellow dot is found by calculating the closest intersection point of these two lines. One can also refine the location of the point by creating more virtual lines. If parallel lines are chosen by mistake, the interface allows the user to delete it.

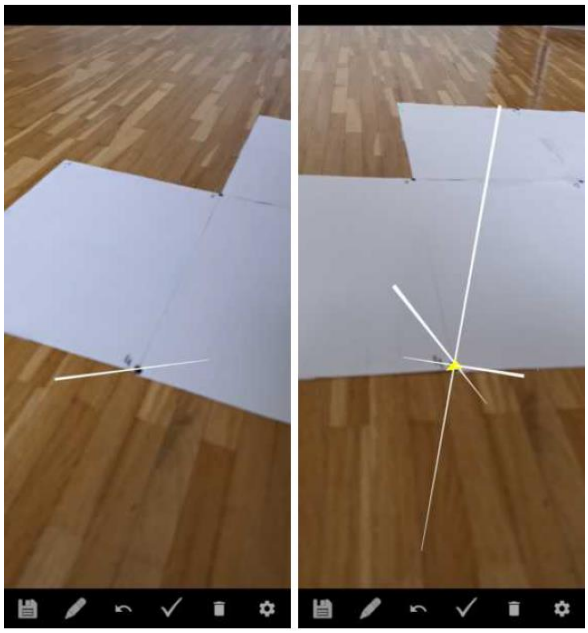

Figure 2. Selection of a point

3.2 VISUALIZATION OF TRAJECTORIES

Learning or programming new trajectories typically require the user to visualize the trajectories before execution on the robot, in order to assure the safety of the robot and the user. Simulation software (e.g., Gazebo) or visualization software (e.g., RViz) can be used for this purpose, but for

a safe verification, this approach requires the user to model not only the robot, but also the robot workspace. This is a tedious task that poorly adapts to new environments or changing workspace. The software that we developed in this deliverable provides an AR approach to circumvent this aspect by visualizing the simulated paths (or other features) directly within the robot workspace. Here, the objective is to visualize the trajectories on the screen of the smartphone, where the camera stream (camera on the other side of the device) is rendered, so that the smartphone can be used as a virtual window that can be brought at any location, and allowing virtual elements to be superimposed to the image. Thus, the user can look for any safety violation and select features or options directly on the screen.

We can teach robots a new task by kinesthetic teaching, by recording end-effector trajectory of the robot, in robot coordinate system. In order to visualize this trajectory in the fixed world coordinate system on the smartphone screen with the camera open, we need a calibration step. This calibration step is required as the robot and the smartphone have different coordinate systems. To calibrate, first, we place the gripper of the robot on N positions of its workspace and record these points in the robot coordinate system. For each of the N points, the application is used to select the corresponding point defined in the world coordinate system of the phone. We then apply an orthogonal Procrustes algorithm to find a homogenous transformation matrix that transforms points expressed in the smartphone coordinate system to the robot coordinate system.

Figure 3 shows the visualization of a robot trajectory that was previously recorded by kinesthetic teaching. It is rendered progressively on the screen as an animation, with the points' sequence corresponding to the movement recorded with the robot.

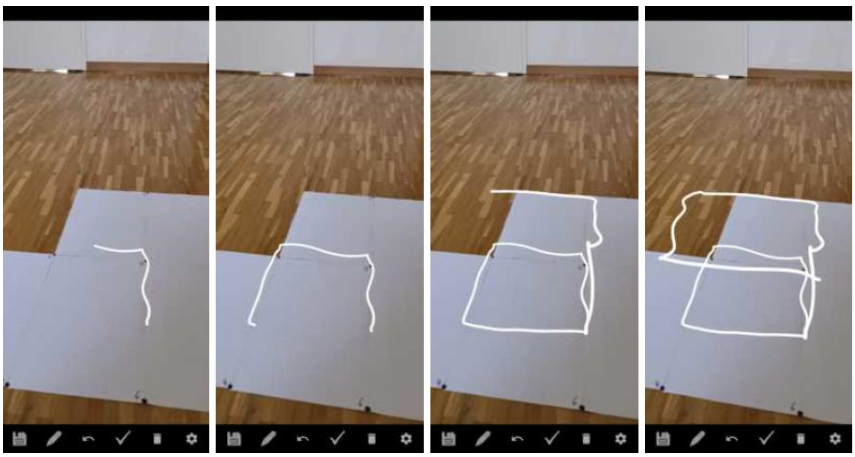

Figure 3. Visualization of a robot trajectory

3.3 VISUALIZATION OF THE ROBOT

Visualization of the robot itself provides an effective communication between the user and the robot. Displaying trajectories on the screen of the smartphone is useful for the user to check for any collisions with the gripper and the environment, uncomfortable distances to other machines or workers, or success of the task demonstrated. However, the user cannot check these criteria for the rest of the robot other than the gripper. In the case of an AGV, when the paths that AGV takes are visualized on the screen, the user can only check optimality of the path and scarce use of resources, but not any safety violations. These can be enforced by a virtual display of the robot on the screen, executing the desired trajectory. When the whole-body of the robot is visualized

with the movements of all joints, success and failures can be surveyed rigorously. In adaptive Human-Robot Collaboration tasks such as the task "T6.5: Challenge 3: Performing LCD TV Assembly" of the project, the intent of the robot can be understood more clearly by the user, when the robot is first displayed on the screen of the smartphone.

We developed a software based on "android-urdfviewer" to visualize a virtual robot in 3D. This Android library implements a ROS node on the Android device to parse the URDF file of the robot and renders the 3D visualization using OpenGL. The URDF parsing and rendering code is a modified version of the corresponding classes from the Rviz for Android project [7], originally released under the Apache License Version 2.0. The necessary dependencies are automatically downloaded by *Gradle* (the build system used by *Android Studio*). This library needs a running ROS master to connect to, and a node that publishes the transforms of the robot on the topics `/tf` and `/tf_static`. Note that this library is still based on the *kinetic* version of ROS.

Figure 4 shows an example implementation of the "visualization of the robot" functionality of our software. We have two real robots in the workspace, and we display a virtual robot with an offset for an effective visualization. This virtual robot replicates the motion of the robot on the left.

Figure 4. Visualization of the robot

4 DISCUSSIONS AND FUTURE WORK

The software presented in this deliverable can be used within a wide range of case studies, to visualize or select trajectories or items in the robot workspace. It provides an inexpensive solution in which any smartphone compatible with ARCore can use the developed application, with a very short training phase. It provides an easy and intuitive interface that we plan to exploit in the next phase to command the robot to do specific tasks, and to visualize the resulting paths, so that the operators can check potential safety violations before running the task on the robot. In the next phase, we will investigate the use cases in which this technology can be showcased.

The selection of points in the workspace of the robot could also be used in connection object recognition algorithms. We, for example, plan to investigate how the user could select an object by clicking on the screen of the smartphone, to choose an object recognized by methods developed within WP4. The user can then specify a point on this specific object for the robot to perform actions. In the task "T6.6: Challenge 4: Collaborative riveting for aircraft parts assembly", the workers can select different parts of the plane to work on only by clicking on the screen. We will investigate whether this could eliminate the need for any small-scale programming and provide an easy interface that can be used by non-experts as well.

Errors in the selection process would cause discrepancies in the displayed trajectories and in the potential teaching and learning phase of the robot as the calibration step requires careful selection of the points. The main errors of the calibration come from the non-accurate and non-precise selection of the calibration points. We will investigate how this calibration step can be optimized using sensory information provided from the WP4, by exploiting the cameras placed in the environment of the robot manipulator and the AGV. This would provide an automatic and very intuitive calibration step. A possible improvement would be to display in real-time on the phone's screen, the error concerning every intersection point, to inform the user about a possible incorrectly placed intersection point (for example by changing the color of the point if the error is above a given threshold), and to be able to only correct the positions of the problematic intersection points.

The software developed until M12 and presented in this deliverable contains preliminary functionalities, currently as separate software. These initial developments set up the stage for the upcoming teaching /learning phases. The aim is to exploit these AR functionalities together with trajectory learning methods to visualize not only position sequences but also variations in the paths. We will next investigate how the behavior representation model provided by WP3 can be exploited to visualize these variations in the trajectories.

REFERENCES

[1] G. Michalos, P. Karagiannis, S. Makris, Ö. Tokçalar and G. Chryssolouris, "Augmented reality (AR) applications for supporting human-robot interactive cooperation," in *Procedia CIRP*, 2016.

[2] C. Mateo, A. Brunete, E. Gambao and M. Hernando, "Hammer: An Android Based Application for End-User Industrial Robot Programming," in *IEEE/ASME 10th International Conference on Mechatronic and Embedded Systems and Applications (MESA)*, 2014.

[3] Google, "ARCore supported devices," [Online]. Available: https://developers.google.com/ar/discover/supported-devices. [Accessed 25 09 2019].

[4] Google, "ARCore," [Online]. Available: https://developers.google.com/ar. [Accessed 05 29 2019].

[5] "Huawei Mate20 Pro," [Online]. Available: https://consumer.huawei.com/en/phones/mate20-pro/. [Accessed 25 09 2019].

[6] "FRANKA EMIKA," [Online]. Available: https://www.franka.de/. [Accessed 26 09 2019].

[7] "Rviz for Android," [Online]. Available: http://wiki.ros.org/rviz_for_android. [Accessed 25 09 2019].