Interactive Robots Control Using Mixed Reality

Article *in* IFAC-PapersOnLine · January 2019

DOI: 10.1016/j.ifacol.2019.11.307

3 authors:

Mikhail Ostanin
Innopolis University

Rauf Yagfarov
Innopolis University

Alexandr Klimchik
Innopolis University

Available online at www.sciencedirect.com

ScienceDirect

IFAC PapersOnLine 52-13 (2019) 695–700

Interactive Robots Control Using Mixed Reality ⋆

M. Ostanin * **R. Yagfarov** * **A. Klimchik** *

* *Innopolis University, Innopolis, 420500, Russia (e-mail: m.ostanin, r.yagfarov, a.klimchik@innopolis.ru).*

Abstract: The paper presents a mixed reality-based approach for interactive control of robotic manipulators and mobile platforms. In particular, we designed an interactive and understandable interface for human-robot interaction. The interface provides tools for robot path programming and visualizes it. Path visualization helps workers understand robot behavior, it is important for safety human-robot interaction. The paper presents an architecture of that system and the implementation for an industrial robot KUKA iiwa and mobile robot platform Plato. The main issue of a multi-platform system is related to the synchronization of coordinate frames for all elements. To deal with this problem we realized 3 setting options: manually, by a camera with markers, point clouds processing. We implemented our interface on Microsoft HoloLens and evaluated it on users.

Keywords: Human-robot interaction, Mixed reality, Augmented reality, Industrial robots, Mobile robots, HoloLens

1. INTRODUCTION

Robots succeed in performing precise, fast, repetitive tasks and have no fatigue, that are useful in manufacturing. When robots are working with humans in the same workspace, it is expected that robots will behave predictably, safely and will complete their task even in the case of unpredictable human behavior. Although programmed robot movements are determined in advance by trajectory planning, it is difficult to transfer these movements to a person. In practice, human-robot interaction (HRI) requires intuitive and efficient interfaces Fong et al. (2003). One of the best ways to implements this in HRI is utilization Augmented and Mixed reality Goodrich et al. (2008).

Mixed Reality (MR) and Augmented Reality (AR) create some space, where both physical and virtual elements can exist together. In the AR digital image is superimposed on the real world, the user does not feel the virtual object as real. MR gives the user the feeling of the hologram really presenting that makes the interface more intuitive and attractive. In addition, the flexible connection of virtual and real environments allows elements in one world react directly to others through direct data transfer. In industry such systems can improve HRI processes, people will see the trajectory of the robot motion and send commands to him interactively. Such systems will increase safety, speed up the programming and reduce the training time for new employees since the interface looks clear and intuitive.

As it was mentioned above, MR reality is the incorporating of the real and virtual worlds, robots and humans should operate in one coordinate system. This is necessary for sending commands to the robot accurately and receiving feedback. The system precision will directly depend on the accuracy of achieving a unified coordinate system. Each robot has its own coordinate system and reference to real space. Industrial robots are fixed in place, but mobile robots are free and localize itself using sensors, such as lidar and camera. Human is localized by the sensors on MR glasses, particularly, it is depth cameras in HoloLens. The compilation of all system elements into one coordinate space is the main problem when creating a system for controlling robots through MR. Moreover, an important aspect of the system is the interface, whereby a person controls the robot and receives feedback, it should be universal for different types of robots, but at the same time adapted to the specifics of the robots, in our case for industrial and mobile robots.

In our research, we are developing a system for controlling various types of robots: the industrial collaborative robot KUKA iiwa and the mobile platform Plato, using the Microsoft HoloLens. All system objects are connected via ROS Kinetic. The scientific contribution of our work consists in the following: a description of the typical system architecture, the creation of algorithms for combining all the elements to one world coordinate system, an interaction interface with the same structure for different types of robots and testing the system on users.

2. RELATED WORKS

Augmented Reality technology is used for adding digital objects to a real environment, in order to create various impressions or improve perception. Over the past few years, various attempts have been made to apply AR technologies in robotics. For example, the AR system was developed to identify virtual obstacles, determine tool positions and assign robots Gaschler et al. (2014). The AR visualization interface was introduced in Livatino et al. (2012) for the

⋆ The work presented in this paper was supported by the grant of Russian Science Foundation 17-19-01740.

2405-8963 © 2019, IFAC (International Federation of Automatic Control) Hosting by Elsevier Ltd. All rights reserved.
Peer review under responsibility of International Federation of Automatic Control.
10.1016/j.ifacol.2019.11.307

simultaneous presentation of information about visual and laser sensors, further enhanced by stereoscopic viewing and three-dimensional graphics. In the papers Fang et al. (2009, 2012) the system on the interactive programming of robots by means of AR is presented, a more advanced similar system based on MR is presented in Ostanin and Klimchik (2018).

The original Mixed Reality definition is presented in Milgram and Kishino (1994). The main advantages of MR for robotics were highlighted in Hoenig et al. (2015), they include space flexibility, security, simplification of debugging, unlimited modification and shared space. All these advantages make MR systems very useful and actual for robotics. The systems for simulation and programming robots based on MR were presented in Ostanin and Klimchik (2018); Wassermann et al. (2018), where the authors focused on the interaction with the industrial robots. Programming organizes by arranging virtual control points in the robot workspace and displaying the trajectory through Microsoft HoloLens. In addition to interactive programming, researches focused on the robots teleoperation. For example, in Cousins et al. (2017) a system based on Oculus Rift DK2 glasses and a LEAP Motion controller of hand movements is presented, with the help of these devices a person controls a Baxter robot.In addition, there are studies aimed at analyzing MR interfaces Rosen et al. (2017); Walker et al. (2018), which describe its potential for robotics and provide an assessment of real users.

To connect a robot and a person to one coordinate system, the above works often use markers that are detected by cameras on devices. In addition, we will evaluate your interface on real users. This is not a robust method, in our work we present another method based on sensor fusion between robot and MR grasses.

3. SYSTEM ARCHITECTURE AND IMPLEMENTATION

3.1 System Overview

The system presented in our work can be divided into three main parts: Interface, Controller and Robots. However each robots may exist in 2 environments: Virtual and Real. A robot can exist in both different environments or only in the one of them. The MR interface and MR device, like Microsoft HoloLens, connect that 2 worlds together. Control unit is connected to each part. Controller realize calculations, transferring commands between the interface and robots, motions of holograms and robots control. Block schema of this division is shown in Fig. 1.

Detailed system block diagram is given in Fig. 2. The figure shows three parts: HoloLens, ROS and Robots. HoloLens represents the MR Interface and Virtual Environment. ROS represents the Controller, that segment consist of "Point clouds matcher", Plato and iiwa controllers. We implement our system based on ROS Kinetic. "Spatial mapping" and "Velodyne VLP-16" blocks send point clouds to "Point clouds matcher" which calculate a transformation between them. "Spatial mapping" represents the HoloLens environment and Velodyne - real environment. From HoloLens Interface block commands sends to robots through "Coordinates transformation" using "Trajectories

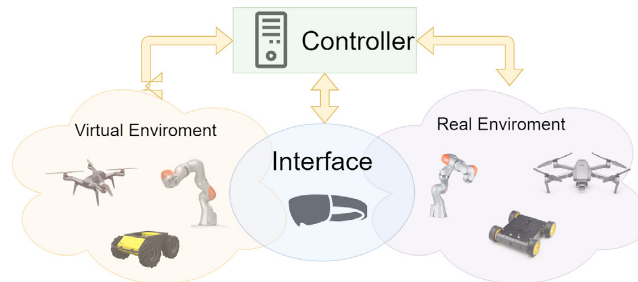


Fig. 1. Main concept of the system

controller". Plato have software blocks for mapping, localization and control. Iiwa controller has inverse Kinematic solver based on SNS algorithm Flacco et al. (2015). On robot board we have sensors, actuators and software with libraries for connection to ROS.

Main Pipeline of the system:

(1) Initialization and matching of the coordinate systems: the mobile platform and HoloLens send clouds of points to the Point clouds matcher node, which calculates the transformation from one coordinate system to another. Then, using marker recognition, the transformation from the HoloLens to the robotic arm coordinate system is calculated. After the step of comparing systems of axis, all elements can operate in one common space.
(2) Using the interface, one pose or a sequence of pose are established, and then are transmitted to the Control unit.
(3) In the control unit, the main calculations are performed: for a mobile robot construction of a map and trajectory planning are realized, and for a robotic manipulator inverse kinematics and trajectory planning problems are solved.
(4) The control unit sends control commands to robots and movements are executed.

3.2 Hardware and Software

HoloLens Microsoft HoloLens is a Mixed Reality glasses. They base on the UWP (Universal Windows Platform) and we implemented the application for it. We used game engine Unity 2017.4 and MRTK (Mixed Reality Toolkit). This toolkit includes basic scripts and examples for the accelerated creation of applications for HoloLens. For connection to ROS, we used RosSharp [1] for UWP, which we modified by adding new message types that are used in ROS.

Industrial robot Our system can work with various industrial robotic manipulators such as KUKA iiwa, KUKA Agilus, Fanuc etc. For the experiments, we chose the lightweight collaborative robot KUKA iiwa LBR 14. This robot is equipped with force-torque sensors and can respond to human touch. Sunrise Workbench is responsible for managing this robot, we use a ready implementation of the robot connecting to ROS Kinetic, it ROS_Smartservo from iiwa_stack [2]. The control is performed by sending the target joint position to the robot controller.

[1] https://github.com/siemens/ros-sharp
[2] https://github.com/IFL-CAMP/iiwa$_s tack$

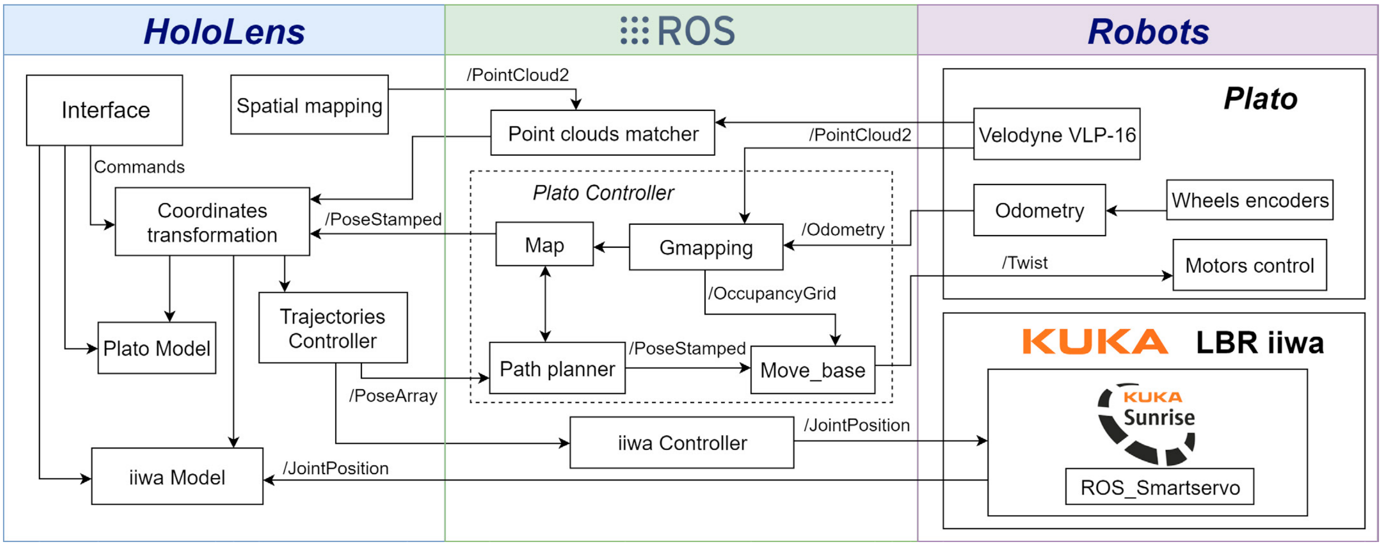


Fig. 2. Mixed reality based robots control system architecture

Mobile robot platform

All our experiments related with mobile robots control were performed on a differential drive mobile robot platform called Plato. Plato is equipped with the onboard computer Jetson TX1, bumper sensors, sonars, Troyka IMU module, Velodyne VLP-16 3D Lidar, RGB camera and wheel encoders that allow to calculate the wheel odometry. Wheel motors are controlled by Arduino Mega. The robots onboard computer uses Ubuntu 16.04 installed with ROS Kinetic.

To navigate a mobile wheeled robot, it need to localize it in space and build a 2D space map. The algorithms that solve this kind of problem are called SLAM (Simultaneous Localization And Mapping). For our robot, we used the Gmapping algorithm, which proved to be the best for our scenarios in comparison with other popular algorithms available in the ROS such as Cartographer Hess et al. (2016) and Hector Slam Kohlbrecher et al. (2011). Comparison of these algorithms has been shown in Yagfarov et al. (2018). Gmapping is based on Rao-Blackwellized particle filter Grisetti et al. (2007). It takes raw laser range data and odometry as input and outputs a map in occupancy grid format. Occupancy grid consists of cells that correspond to the state in the physical space and they can be in one of two states: free or occupied. After the robot has an environment map, it is necessary to build a motion trajectory. For this we used the move_base ROS package, which takes point in 2D space and starts global and local planning. Move_base node also makes two cost maps: global and local.

3.3 World Coordinate System

An important part of our system is solving the problem of comparing coordinate systems in order to work in the single coordinate space. This problem can be solved in several ways: setting the origin manually, detecting the origin with the help of the markers (QR, Aruco, etc) and by comparing the clouds of points obtained with HoloLens to the points obtained with the 3D lidar Velodyne VLP-16. In the first case, the user places the origin of coordinates in the place in the space where the mobile robotic platform is located using HoloLens. In the case of camera markers, the same label is detected using cameras on HoloLens and on the mobile platform and thus this label is the link between the components of the system. Third method is the most flexible and convenient for the user, where coordinate systems are automatically determine relative positions of their origins and do not require any additional actions such as printing labels, etc.

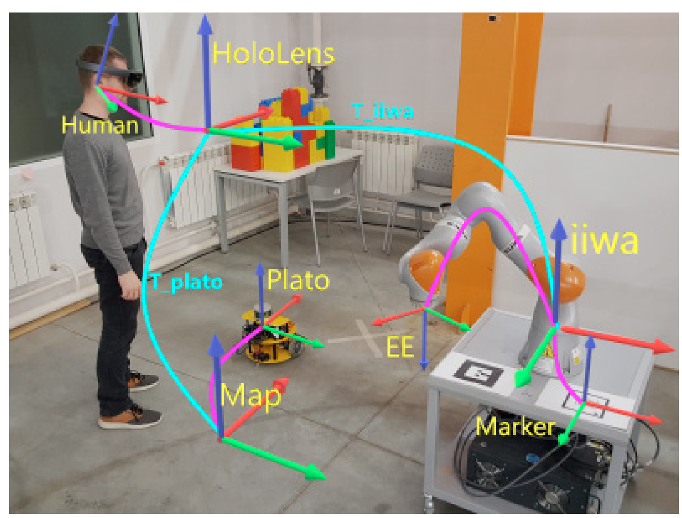


Fig. 3. Principal frames and transformations of the system

Now let us present all principal frames of the system and transformations between them. Figure 3 shows principal components of the system used further in this work. HoloLens initialize T_HoloLens coordinate system when launching the application, it is located in started position. Then virtual objects have coordinates relative to T_HoloLens. Frame T_Human describes the position and orientation of the HoloLens (user). KUKA iiwa is fixed in space. Manipulator position is recognized by a marker that is fixed relating to the robot base. The camera on the HoloLens detects the marker, thereby achieving the transformation T_iiwa through Marker. The mobile platform has its own coordinate system for mapping and localization. For the connection between the Plato and HoloLens, we match two point clouds. The first one is a lidar frame relative to T_Map and the second one is Spatial Mapping to T_HoloLens. As a result, we obtain a transformation T_plato. Thus, we gain all the necessary transformations in our system.

Comparison of coordinate systems is based on the comparison of point clouds and occurs in several stages:

(1) HoloLens scans the room and builds a cloud of points.
(2) Using 3D LIDAR on board the mobile platform, we get a cloud of points from one scan.
(3) Using modified ICP (Iterative Closest Point) algorithm Besl and McKay (1992), we compare clouds of points and obtain the transformation matrix of one coordinate system to another. Algorithm 1 shows the process in detail.

Algorithm 1 Point clouds matching algorithm

Input: P - point cloud from Plato robot,
H - point cloud from HoloLens,
n_iter - number of iterations
ϵ - matching tolerance
Output: T - transformation from P to H origin
$min\_n\_points = \min(size(P), size(H))$
if $min\_n\_points ==$ size(P) **then**
  $H =$ select_random_elements(H, min_n_points)
else
  $P =$ select_random_elements(P, min_n_points)
end if
$cm\_P =$ center_of_mass(P)
$cm\_H =$ center_of_mass(H)
match(cm_P, cm_H)
while $error > \epsilon$ and $i < n\_iter$ **do**
  for each $p_i \in P$ **do**
    $indices_i =$nearest_point(p_i, H)
  end for
  $T, error = \min_T \sum_i ||h_i - T(p(indices_i))||^2$
    =least_squares$(P, H, indices)$
  $T =$ only_yaw_rotation(T)
  $P =$ transform(P, T)
  $i++$
end while=0

The rotation of the point cloud was performed relative only to the z axis (yaw), since the z-axe unit vectors of given point clouds coordinate frames are collinear and perpendicular to the floor surface. This approach reduces the chance that ICP will stop at a local minimum.

4. MIXED REALITY INTERFACE

Mixed reality (MR) combines a real and virtual environment and creates a connection between real objects and holograms. In the spectrum from the real to the virtual world, MR is in the middle, it includes the capabilities of augmented and virtual reality. For immersion in MR, we use Microsoft HoloLens. The interaction is based on hand gestures and a virtual cursor, which is located in the display center and is controlled by turning the head. The glasses recognize the wrist position and the main gesture

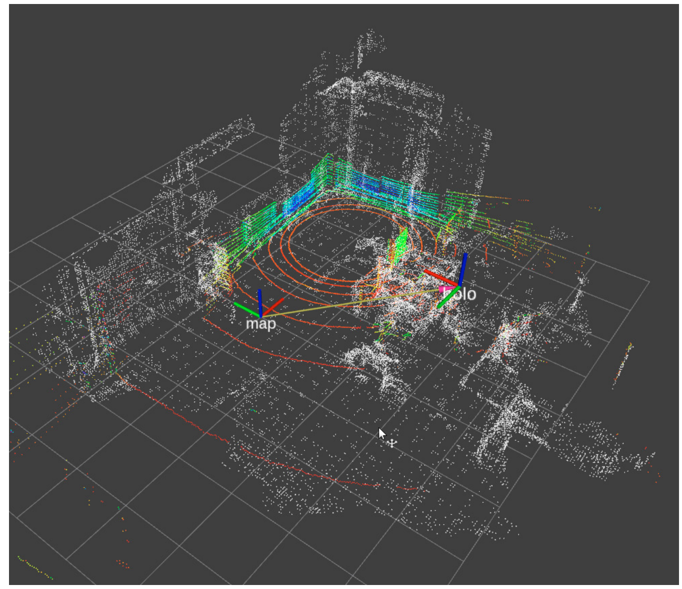


Fig. 4. Result of point cloud matching. Colored points are points from Velodyne scan, and white points are points from pointcloud created by HoloLens. "map" and "holo" are origins of two frames.

- air click. These opportunities will be the basis of our interface.

When the application is launched, the user is presented with a virtual menu that contains options to set up and control robots. The menu consists of 2 main tabs, each corresponding to its robot iiwa and PLATO. Each tab consists of the same set of commands.

- Set Up. Installation of a virtual robot model in place of a real one.
- Add Point. Adding a control point that describes the position to which the robot should move.
- Delete Point. Deleting the last point in the path.
- Send. Sending the path to the robot.

Each robot has its own virtual model (Fig. 5a, 5b). We use a variety of technical methods to place virtual robot models in real ones, by markers for the manipulator and by processing point clouds for the mobile platform.

The script using the interface is working as follows:

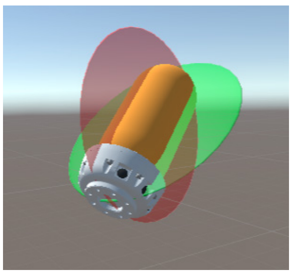
(1) Choosing a robot
(2) Initialization
(3) Add a point / delete point
(4) Modification points
(5) Send the trajectory

Modification of the points is possible at any time and the robot's path is automatically rebuilt.

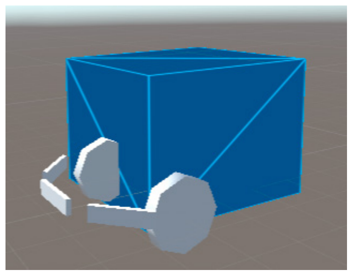
The view and setup process of control points is different for our robots since they have different degrees of freedom(DOF). The mobile platform has 3 DOF, in Cartesian space, it's the XY position on the floor and the orientation around Z. For the manipulator end-effector (EE), we set all 6DOF, completely position and orientation.To visualize the goal, we use a robot model for Plato and an EE model for iiwa (Fig. 5c, 5d).

(a) Iiwa virtual model.

(b) Plato virtual model.



(c) Iiwa goal.



(d) Plato goal.

Fig. 5. Virtual objects.

HoloLens contains an environment - spatial map, therefore setting a control point for the mobile platform is just a click on the floor. The cursor indicate the position, and the rotation of the user body sets the orientation. To set the manipulator points, there are 2 options. The first is manually changing the position and orientation. The position is set by clicking on the point model and moving it by hand - 3DOF. By clicking on the circles and moving the hand to left and right, the user changes the model orientation on that plane. The second option is to click on the object surface in the robot workspace. The cursor is set the pose with orientation along the surface normal, it remains to set only the orientation around the normal axis.

The user can control robots directly, that means setting one target pose for the KUKA iiwa or PLATO and send it. We use a robots model and EE model For setting PLATO and iiwa goal, respectively. A more advanced robot control option is the programming, in other words, to design a robot path. For a mobile platform, this is the path on the floor which is specified by the sequence of control positions. The robotic arm path is the poses in Cartesian space, which are presented by the sequence of control points models. The motion trajectory between points is drawn by a straight line, this helps to understand points connection (Fig. 6).

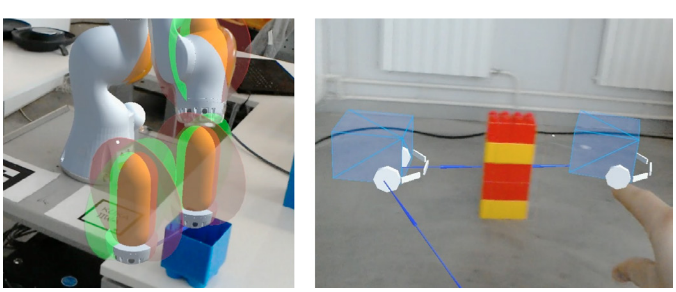


Fig. 6. Robots paths example.

5. USER STUDY

In order to evaluate the system The study was divided into several stages. At the first stage, the training was performed in which we explain the interface of our system for each participant. Also, the system capabilities were shown on the example tasks. Subjects were asked to specify one goal point and move the robot into it, and then a sequence of goal points and conduct the robot through them. At the second stage, the user was given tasks for each robot separately. For the robot Plato it was parking the robot in the garage and avoid the obstacles along the eight-shaped trajectory, and for the robot KUKA iiwa it was a hit by the EE inside a small box and avoiding obstacles made from the blocks. At the last stage, the user was given a group task, in which both robots were to be involved. The group task in our experiments was the delivery of the object lying in point A on the table to point B. Formally, the task can be divided into several sub-tasks:

(1) Bring Plato robot to point A near the table
(2) Take the object lying on the table using a robotic manipulator Kuka iiwa
(3) Placing an object on Plato
(4) Deliver the object to point B.

In our experiments, it was important to select problems that were not highly dependent on specific human skills and were sufficiently general, so the task that we asked the subjects was fairly simple and consistent, but our system can be adapted to more dynamic tasks and conditions. Despite the simplicity of the tasks and not their dependence on professional skills, one of the main goals of our research was to find out how easy a new user will be able to use our system. For this purpose, people who never worked with the stack of technologies presented in our work, and people working with these technologies daily and on a professional level participated in the experiments.

5.1 Procedure

Spending time on performing task was recorded. To obtain the numerical rating of the system, a questionnaire was developed in which the user could evaluate the individual components of the system and give an estimate of 1 to 5. Numerical data obtained on the basis of answers in the questionnaires are presented in Table 1. Also, all the experiments were recorded on video for further analysis and measuring the execution time of each stage.

5.2 Results

To understand how our system is simple, intuitive and accurate, a user study was conducted. For this purpose, 23 participants of the experiment were invited, among them 8 beginners, 15 professionals. The age of the subjects is between 19 and 36, the average age is 25, of which 17 are males and 6 are females. All users complete with the task with different time. The mean time for complete study task is 11 min. with standard deviation equal to 5.5 min. For completing the test task participants spend approximately 11 ± 4 minutes. Results of the questionnaire are shown in Table 1.

Table 1.

Question	Mean	std
Did you find interface based on Mixed Reality for robot control simple?	4.00	0.85
Did you find interface based on Mixed Reality for robot control intuitive?	4.04	0.93
How would you rate your result? Was robot motion precise?	3.7	0.82
Did you find interface based on Mixed Reality for robot control helpful?	4.7	0.70
Did you find Mixed Reality interface convenient to control robot?	4.26	0.75
Did you find interface based on Mixed Reality for robot control easy to explain how to work with?	4.39	0.78

6. CONCLUSION AND FUTURE WORK

The article presents a system for interactive programming of different robot types based on mixed reality. A typical architecture consisting of 3 main parts is described: interface, controller, robots. The interface was implemented on Microsoft HoloLens mixed reality glasses. The controller is based on Kinetic ROS. For the experiments, the PLATO mobile platform and the KUKA iiwa manipulator were chosen. Various algorithms for creating a unified world coordinate system have been developed. For industrial robots, we used a marker-based method and for the mobile platform, we realized the sensor fusion algorithm between HoloLens point cloud and lidar. The study showed that the proposed system is helpful, intuitive and convenient but not very precise. The accuracy of our system directly depends on the devices which we used. The average studying time is 11 minutes, that shows how easy to teach work with our system.

In the future, we plan to extend our system in 2 main directions. The first direction is adding new robot types such as bipedal robots and drones. The second goal is to create interfaces used another type of digital reality, i.e. augmented and virtual reality (AR and VR). AR will be like web interface with cameras steam augmented by holograms. VR view will be based on VR headsets and will visualize only the virtual environment. As a result, we want to archive a multi-view system for programming and control multi-robots.

ACKNOWLEDGMENT

The work presented in this paper was supported by the grant of Russian Science Foundation 17-19-01740.

REFERENCES

Besl, P.J. and McKay, N.D. (1992). Method for registration of 3-d shapes. In *Sensor Fusion IV: Control Paradigms and Data Structures*, volume 1611, 586–607. International Society for Optics and Photonics.

Cousins, M., Yang, C., Chen, J., He, W., and Ju, Z. (2017). Development of a mixed reality based interface for human robot interaciotn. *2017 International Conference on Machine Learning and Cybernetics (ICMLC)*, 27–34.

Fang, H., Ong, S., and Nee, A. (2012). Robot Path and End-Effector Orientation Planning Using Augmented Reality. *Procedia CIRP*, 3, 191–196.

Fang, H., Ong, S.K., and Nee, A.Y.C. (2009). Robot Programming Using Augmented Reality. *2009 International Conference on CyberWorlds*, 13–20.

Flacco, F., De Luca, A., and Khatib, O. (2015). Control of redundant robots under hard joint constraints: Saturation in the null space. *IEEE Transactions on Robotics*, 31(3), 637–654.

Fong, T., Nourbakhsh, I., and Dautenhahn, K. (2003). A survey of socially interactive robots. *Robotics and autonomous systems*, 42(3-4), 143–166.

Gaschler, A., Springer, M., Rickert, M., and Knoll, A. (2014). Intuitive robot Tasks with augmented reality and virtual obstacles. *Proceedings - IEEE International Conference on Robotics and Automation*, 6026–6031.

Goodrich, M.A., Schultz, A.C., et al. (2008). Human–robot interaction: a survey. *Foundations and Trends® in Human–Computer Interaction*, 1(3), 203–275.

Grisetti, G., Stachniss, C., and Burgard, W. (2007). Improved techniques for grid mapping with rao-blackwellized particle filters. *IEEE transactions on Robotics*, 23(1), 34–46.

Hess, W., Kohler, D., Rapp, H., and Andor, D. (2016). Real-time loop closure in 2d lidar slam. In *Robotics and Automation (ICRA), 2016 IEEE International Conference on*, 1271–1278. IEEE.

Hoenig, W., Milanes, C., Scaria, L., Phan, T., Bolas, M., and Ayanian, N. (2015). Mixed reality for robotics. 5382–5387.

Kohlbrecher, S., Von Stryk, O., Meyer, J., and Klingauf, U. (2011). A flexible and scalable slam system with full 3d motion estimation. In *Safety, Security, and Rescue Robotics (SSRR), 2011 IEEE International Symposium on*, 155–160. IEEE.

Livatino, S., Banno, F., and Muscato, G. (2012). 3d integration of robot vision and laser data with semi-automatic calibration in augmented reality stereoscopic visual interface. *environments*, 15(18), 19.

Milgram, P. and Kishino, F. (1994). A taxonomy of mixed reality visual displays. *IEICE TRANSACTIONS on Information and Systems*, 77(12), 1321–1329.

Ostanin, M. and Klimchik, A. (2018). Interactive robot programing using mixed reality. *IFAC-PapersOnLine*, 51(22), 50–55.

Rosen, E., Whitney, D., Phillips, E., Chien, G., Tompkin, J., Konidaris, G., and Tellex, S. (2017). Communicating Robot Arm Motion Intent Through Mixed Reality Head-mounted Displays. 1–16.

Walker, M., Hedayati, H., Lee, J., and Szafir, D. (2018). Communicating robot motion intent with augmented reality. 316–324.

Wassermann, J., Vick, A., and Krüger, J. (2018). Intuitive robot programming through environment perception, augmented reality simulation and automated program verification. *Procedia CIRP*, 76, 161–166.

Yagfarov, R., Ivanou, M., and Afanasyev, I. (2018). Map comparison of lidar-based 2d slam algorithms using precise ground truth. In *2018 15th International Conference on Control, Automation, Robotics and Vision (ICARCV)*, 1979–1983. IEEE.