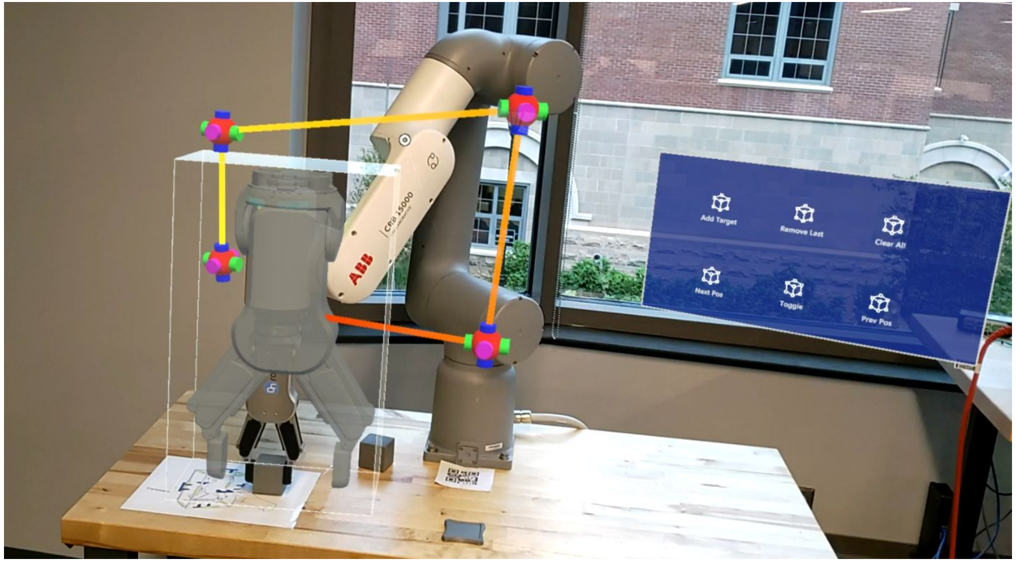
Path Programming for Industrial Robots

Related work, ideas and prototype

What is path programming?

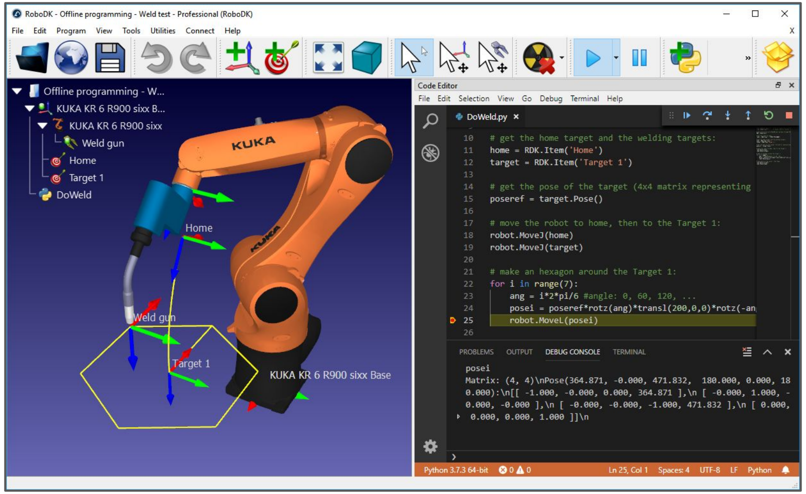
We call path programming the programming method where the user defines instructions for a robot through an interface that is based on a directed graph. Path programming can be an alternative to conventional programming methods found in industrial robots (e.g., lead-through programming, offline programming).



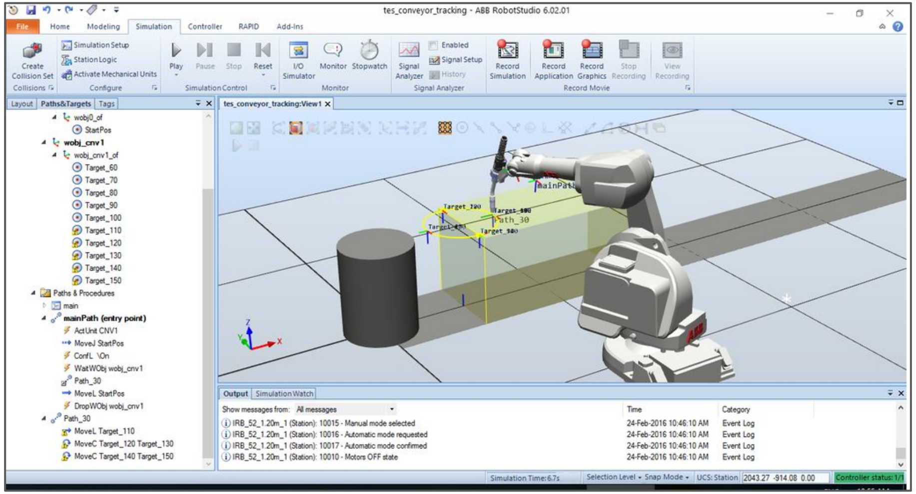
Draft implemented by one of our undergraduate students

Is it something new?

No! The idea of visually representing robot movements through a directed graph is already used by many programming environments such as ABB RobotStudio and RoboDK.



Screenshot from RoboDK

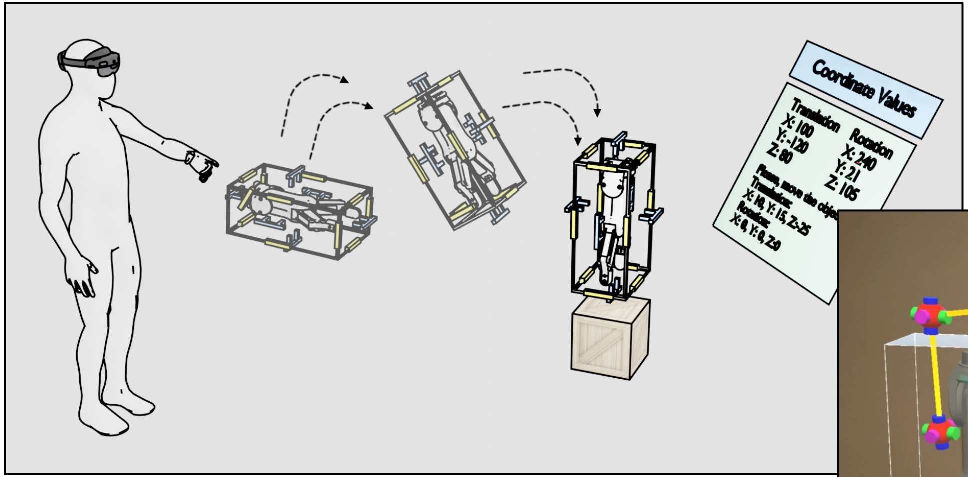


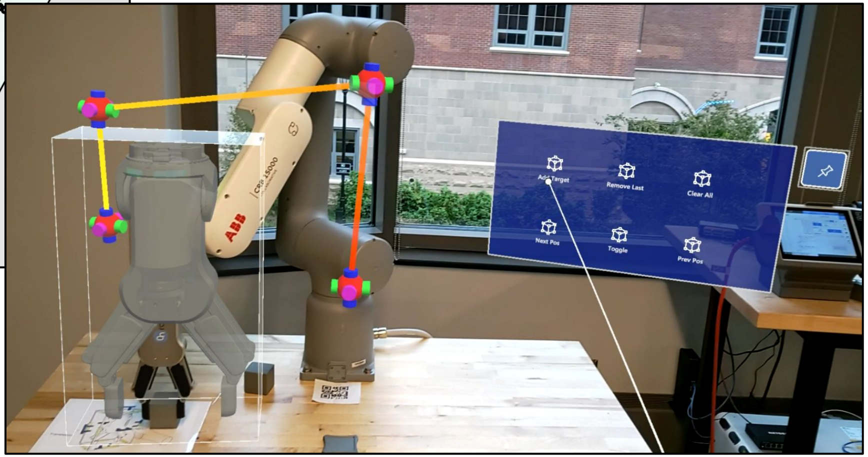
Screenshot from ABB RobotStudio

Image source:
https://pypi.org/project/robodk/
https://us.v-cdn.net/5020483/uploads/editor/pa/1r5vqm5nxqvd.png/

What is our goal then?

Implement a robot path programming environment in mixed reality that is specifically designed for end-users. It should be simple, intuitive and visually appealing. We can combine other recent studies, such as jogging in mixed reality, and create a really powerful programming environment.





What is the state of the art?

We're not the first researchers exploring path programming in mixed reality.

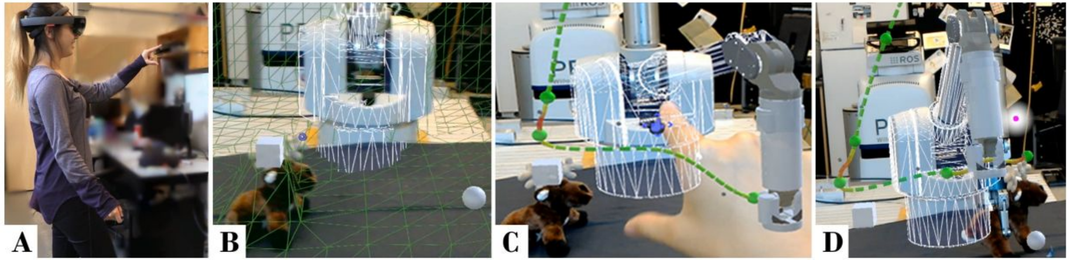


Fig. 3. A pick and place task is completed using our AR Robotics system. (A) An operator wearing the Hololens and gesturing to interact with a 7DOF robot arm. (B) Visualization of, 3D spatial grid, real and virtual robot arm, pick (grey cube) and place (grey sphere) location. (C) Editing visual trajectory by gesturing and then simulate virtual robot through the trajectory. (D) Pick and place execution with virtual and real robot overlapping.

Reference:
Quintero, Camilo Perez, et al. "Robot programming through augmented trajectories in augmented reality." *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018.

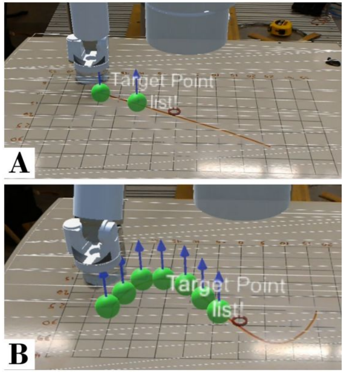


Fig. 6. Participant specifying line (A) and sin (B) shape through the AR-robotic interface.

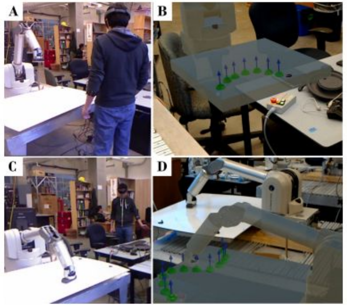


Fig. 4. (A, B) The user set the virtual robot and the CAD model of the dry-erase white board in the side of the real robot and set the path points on the virtual surface. (C, D) The user constrain the end-effector of the robot to the path with an opposite orientation to the surface normals and controls the robot movements inside the path by waving his hand using a MYO device.

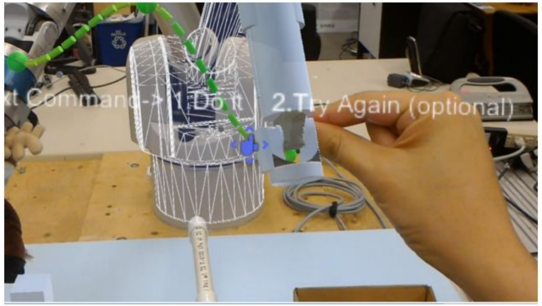


Fig. 7. The participant edits the path to avoid the PVC hurdle and executes a preview to make sure the path is safe.

What is the state of the art?

Most studies are focused on the representation of the direct graph in mixed reality.

Fig. 7. Circle curve. Goal points setting, First arc setting, Simulation.

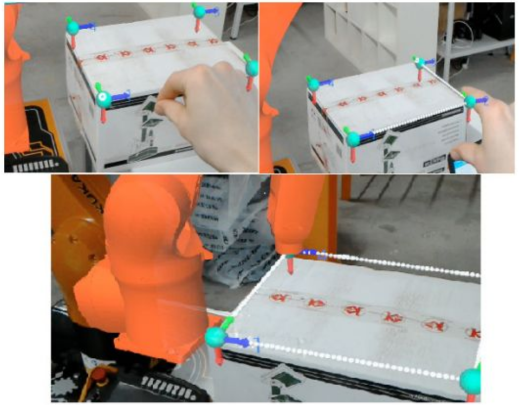


Fig. 8. Rectangle curve. Goal points setting, Lines setting, Path planing.

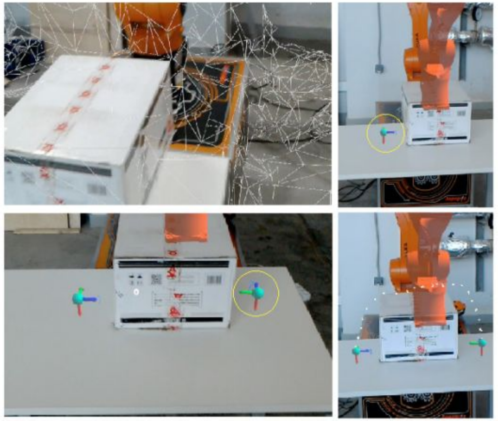


Fig. 6. PTP with collision avoidance. Scanning, Goal points setting, Path planing.

Reference:
Ostanin, Mikhail, and Alexandr Klimchik. "Interactive robot programing using mixed reality." IFAC-PapersOnLine 51.22 (2018): 50-55.

What is the state of the art?

There are also related studies using different technologies for path programming.

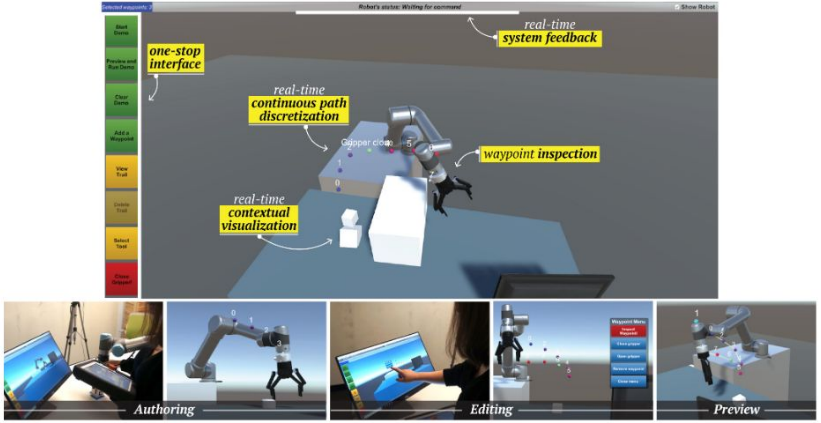


Fig. 3. Demoshop provides users with a one-stop interface through which they can use continuous path discretization to develop and modify programs, mental scaffolds to understand the state of their program, and just-in-time assistance to facilitate their programming process.

Fig. 5. We conducted a between-subjects study in which users were randomly assigned to either a control condition in which they used *Universal Robots'* PolyScope or the experimental condition in which they used Demoshop.

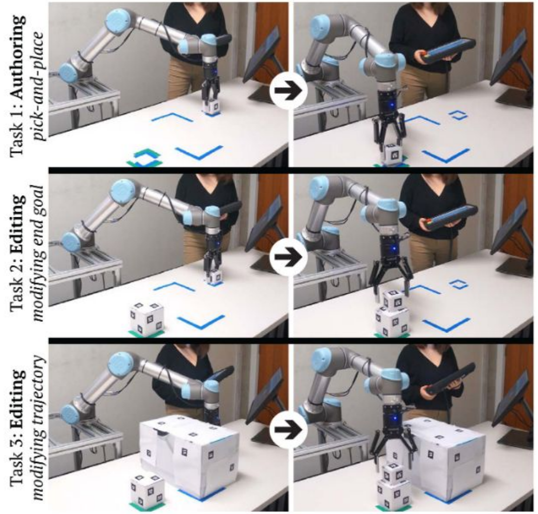


Fig. 4. Our user evaluation consists of three pick-and-place tasks. The first task focused on authoring, while the second and third tasks required the user to edit their existing program to meet new task requirements.

Reference:
Ajaykumar, Gopika, Maia Stiber, and Chien-Ming Huang. "Designing user-centric programming aids for kinesthetic teaching of collaborative robots." Robotics and Autonomous Systems 145 (2021): 103845.

What is the state of the art?

CAD-based environments are a great example of how path programming is important in robotics:

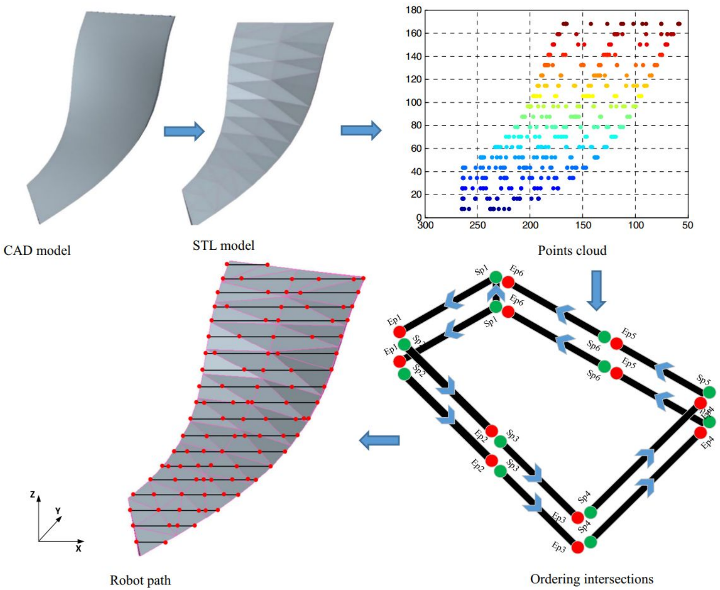


Fig. 10 The process of generating robot programming from CAD

To name a few:

- (Left image) Zheng, Huadong, et al. "CAD-based automatic path generation and optimization for laser cladding robot in additive manufacturing." The International Journal of Advanced Manufacturing Technology 92.9 (2017): 3605-3614.
- Neto, Pedro, J. Norberto Pires, and A. Paulo Moreira. "3D CAD-based robot programming for the SME shop-floor." 20th International Conference on Flexible Automation and Intelligent Manufacturing, FAIM. Vol. 59. 2010.
- Kim, J. Y. "CAD-based automated robot programming in adhesive spray systems for shoe outsoles and uppers." Journal of Robotic Systems 21.11 (2004): 625-634.
- Bedaka, Amit Kumar, and Chyi-Yeu Lin. "CAD-based robot path planning and simulation using OPEN CASCADE." Procedia computer science 133 (2018): 779-785.
- Klein, Alexandr. "CAD-based off-line programming of painting robots." Robotica 5.4 (1987): 267-271.
- Neto, Pedro, et al. "High-level robot programming based on CAD: dealing with unpredictable environments." Industrial Robot: An International Journal (2012).
- Pulkkinen, Topi, et al. "2D CAD based robot programming for processing metal profiles in short series manufacturing." 2008 International Conference on Control, Automation and Systems. IEEE, 2008.

What can we do different?

Limitations identified in past studies:

- Most studies only explore how robots could be moved using directed paths, but do not explore it as an alternative for robot programming, which involves different types of movements, synchronization, gripper manipulation, collision prediction, etc.

- Not all studies clearly specify that their motivations are based on end-users. They don't investigate important aspects such as usability.

- Most studies don’t compare different representations of path programming in mixed reality.

What we could do different:

- Design our prototype as a complete programming environment. Use the knowledge we gained from our questionnaires and apply it to this new experiment to create a beginner-friendly programming alternative.

- Make our study focused on end-users, using as motivation the evolution of collaborative robots and devices such as Hololens. Design an experiment that gets the perspective of end-users on our prototype.

- Create different representations of path programming in mixed reality. Explore what other technologies use to make paths more intuitive and interesting (e.g., Waze).

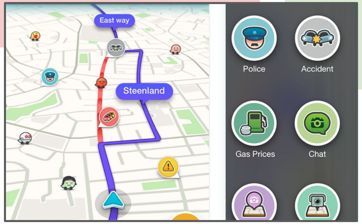


Image source:
https://dynaimage.cdn.cnn.com/cnn/c_fill,g_auto,w_1200,h_675,ar_16:9/https%3A%2F%2Fcdn.cnn.com%2Fcnnnext%2Fdam%2Fassets%2F190207161538-waze-maps.jpg

Week 2

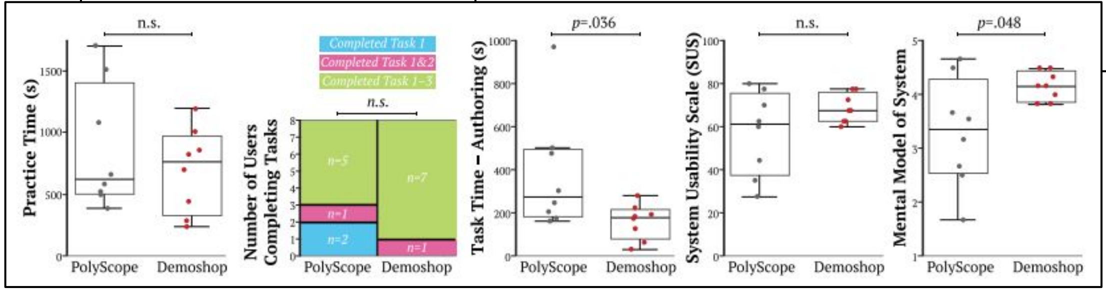
Moving forward the discussion

In-depth analysis of related work

Let’s take a detailed look on other studies on path programming for robots:

Reference: Ajaykumar, Gopika, Maia Stiber, and Chien-Ming Huang. "Designing user-centric programming aids for kinesthetic teaching of collaborative robots." Robotics and Autonomous Systems 145 (2021): 103845.

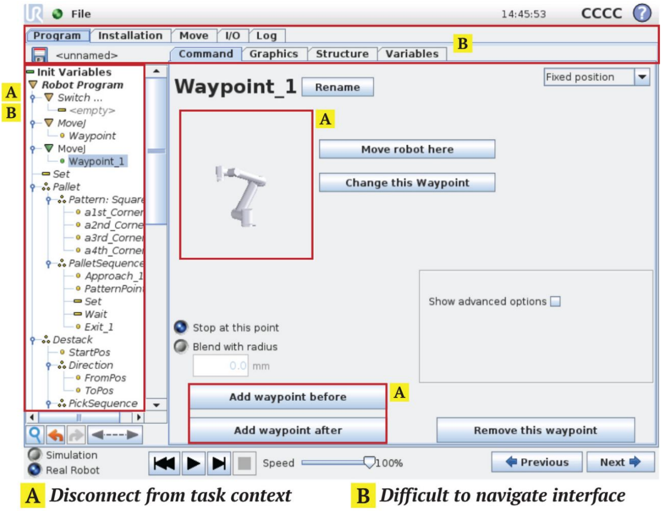
System Description	Method	Evaluation
Demoshop, a Unity desktop application that controls a UR5 robot using ROS.	**1)** Conducted formative study to explore difficulties faced by eight end-users while using the UR PolyScope programming interface. **2)** Implemented a new interface called Demoshop based on the difficulties found in the formative study. **3)** Users were invited to solve three tasks in 50 minutes. 16 participants were divided in two groups: one testing Polyscope, the other testing Demoshop.	**Objective measures:** Practice time (in seconds). Task time (in seconds). Task progress (# tasks completed). **Subjective measures:** SUS Questionnaire Mental Model of System Likert scales regarding system adoption in industry and at home. **Important limitation:** most of their participants had background experience in programming.



In-depth analysis of related work

Let’s take a look on other studies on path programming for robots:

Reference: Ajaykumar, Gopika, Maia Stiber, and Chien-Ming Huang. "Designing user-centric programming aids for kinesthetic teaching of collaborative robots." Robotics and Autonomous Systems 145 (2021): 103845.



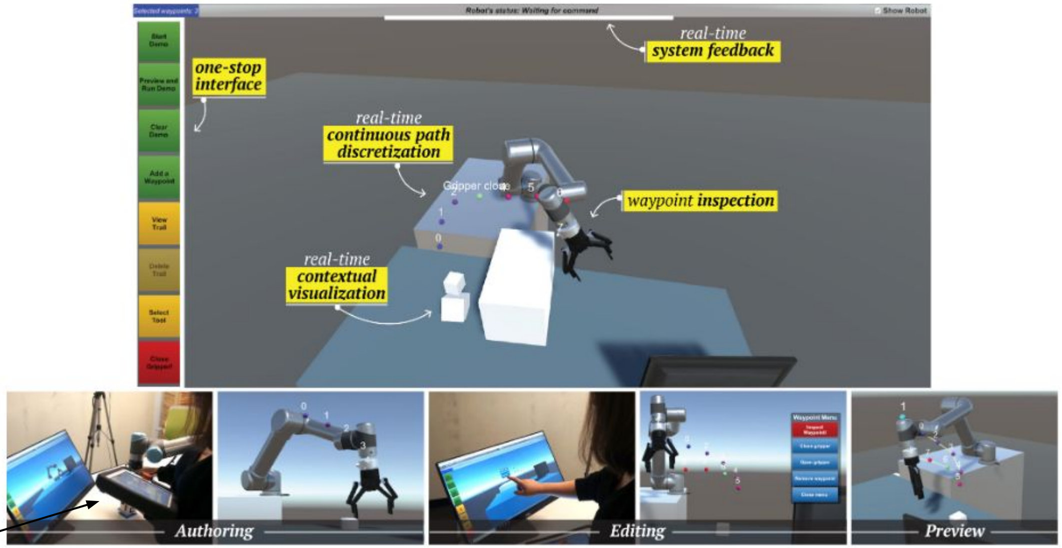


Fig. 3. Demoshop provides users with a one-stop interface through which they can use continuous path discretization to develop and modify programs, mental scaffolds to understand the state of their program, and just-in-time assistance to facilitate their programming process.

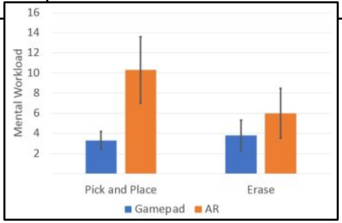
In both strategies, participants used the teach pendant

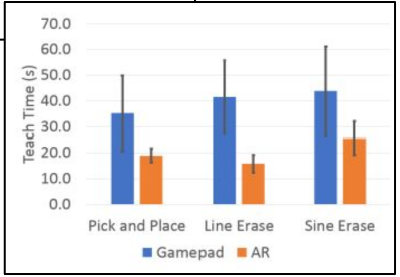
In-depth analysis of related work

Let’s take a detailed look on other studies on path programming for robots:

Reference: Quintero, Camilo Perez, et al. "Robot programming through augmented trajectories in augmented reality." 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2018.

System Description	Method	Evaluation
A Hololens 1 application used to create, edit and visualize paths in Mixed reality. In this prototype, a virtual model of the robot is used as reference, and it is placed next to the real robot. The user manually specifies way points in the mixed reality space from which a path is generated automatically by the system. Using the virtual model, the user can preview, manipulate and update the generated path.	**1)** Description of the prototype. **2)** Execution of pilot study with ten users. Users we divided in two groups: one using the MR prototype, the other using a gamepad. Two tasks were performed by the participants: a) Surface Contact: Participants were asked to erase two different lines, a straight line and a sinusoidal line. b) Free Space: Participants were asked to pick and place an object from a starting position with an obstacle between them. **3)** A use case study was conducted with one participant in a complex real-world manufacturing task. The participant answer a subjective questionnaire after the experiment.	**Experiment:** **Recorded values:** Robot pose, joint trajectories, task completion time and task completeness. **Post-experiment:** NASA-TLX questionnaire. **Case study:** Subjective questionnaire. **Note:** Didn’t specify end-users as their focus.





In-depth analysis of related work

Let’s take a detailed look on other studies on path programming for robots:

Reference: Quintero, Camilo Perez, et al. "Robot programming through augmented trajectories in augmented reality." 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2018.

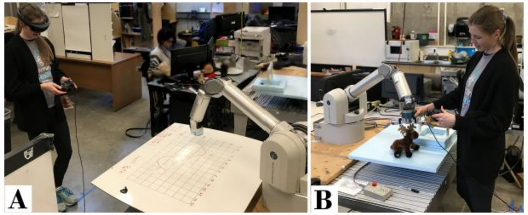


Fig. 5. A) Path specification on a surface setup. B) Path specification on free space setup.

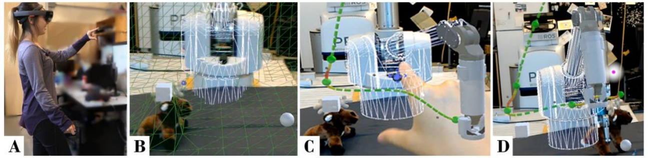


Fig. 3. A pick and place task is completed using our AR Robotics system. (A) An operator wearing the Hololens and gesturing to interact with a 7DOF robot arm. (B) Visualization of, 3D spatial grid, real and virtual robot arm, pick (grey cube) and place (grey sphere) location. (C) Editing visual trajectory by gesturing and then simulate virtual robot through the trajectory. (D) Pick and place execution with virtual and real robot overlapping.

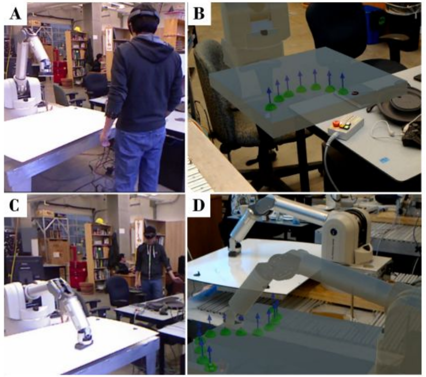


Fig. 4. (A, B) The user set the virtual robot and the CAD model of the dry-erase white board in the side of the real robot and set the path points on the virtual surface. (C, D) The user constrain the end-effector of the robot to the path with an opposite orientation to the surface normals and controls the robot movements inside the path by waving his hand using a MYO device.

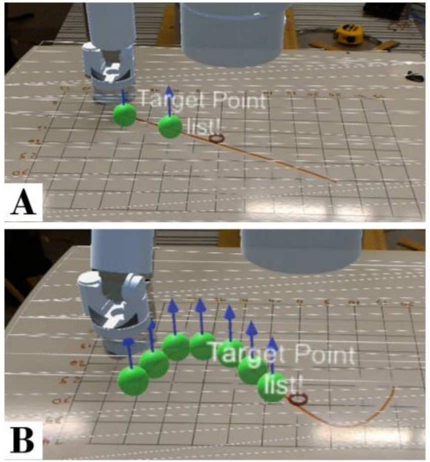


Fig. 6. Participant specifying line (A) and sin (B) shape through the AR-robotic interface.

In-depth analysis of related work

Let’s take a detailed look on other studies on path programming for robots:

Reference: Ostanin, Mikhail, and Alexandr Klimchik. "Interactive robot programing using mixed reality." IFAC-PapersOnLine 51.22 (2018): 50-55.

System Description	Method	Evaluation
A Hololens 1 application built using Unity and the MRTK. The user interaction is carried out through gestures. The path can be generated automatically or implemented manually based on goal points. Automatic path generation includes: point-to-point, line and arc paths. User can also draw the path using a virtual pointer. The application also allows users to scale and update the path.	**1)** Implemented the prototype in Mixed Reality. **2)** Demonstrated the capabilities of the prototype in three different tasks: pick and place, circular trajectory and rectangular trajectory. **3)** Discussed the benefits of Mixed Reality compared to other technologies.	No evaluation was done, just a demonstration.

Table 1. Comparison Digital Reality Features

Feature	**MR**	**AR**	**VR**
Mapping VE and RE	Yes	Yes/No	No
Sharing space	Yes	Only VE	Only VE
Simulation of Virtual in Real Space	Yes	Yes, but without interaction with RE	No
Scale	Yes	Only VE	Only VE
Obstacle avoid-ance	Yes	Additional device to scanning RE	Avoiding of VE
Real Robot Con-trol	Yes	Yes	No

In-depth analysis of related work

Let’s take a detailed look on other studies on path programming for robots:

Reference: Ostanin, Mikhail, and Alexandr Klimchik. "Interactive robot programing using mixed reality." IFAC-PapersOnLine 51.22 (2018): 50-55.

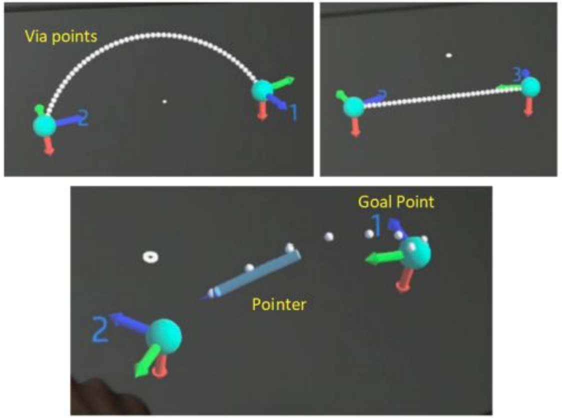


Fig. 4. Lines between 2 goal points: Arc, Line, User path with pointer.


Fig. 7. Circle curve. Goal points setting, First arc setting, Simulation.

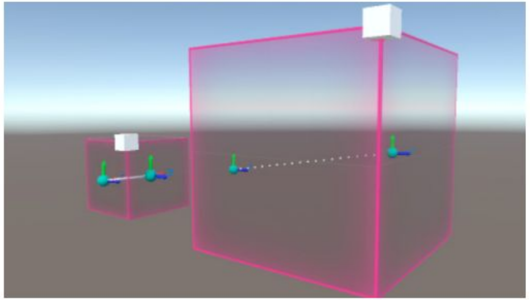


Fig. 5. Path scaling.

In-depth analysis of related work

Let’s take a detailed look on other studies on path programming for robots:

Reference: Piccinelli, Marco, et al. "Trajectory planning using Mixed Reality: an experimental validation." 2021 20th International Conference on Advanced Robotics (ICAR). IEEE, 2021.

System Description	Method	Evaluation
A Hololens 2 application connected to a Panda (Franka Emika) robot using ROS. In the application the user can: Add a new waypoint in a list of waypoints. Play the trajectory using a virtual robot. Send the trajectory to the real robot. Visualize the waypoints and adjust them as necessary. **Interesting feature:** You can adjust the time interval between one point and another.	**1)** Implemented the prototype. **2)** Performed tests on the prototype to check if it works as expected.	Did not run a user study, but plan to do it in the future: “In the future we plan to do an extensive user study to evaluate our system among people with different backgrounds and to compare it with classical methods such as teaching-by-demonstration or the usage of teach pendants. Moreover we will extend the MR also to other cooperative robots as the UR5 by Universal Robot and the KUKA available in our lab.”

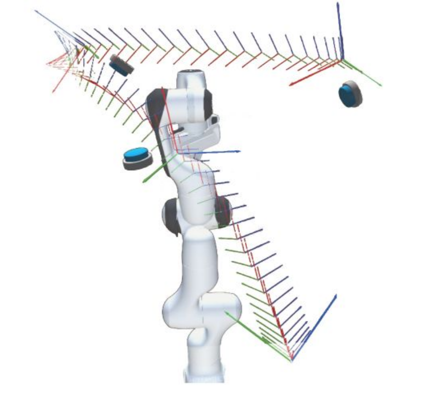


Fig. 2. Holograms representing the designed trajectory, with way points highlighted by their bigger size.

In-depth analysis of related work

Let’s take a detailed look on other studies on path programming for robots:

Reference: Piccinelli, Marco, et al. "Trajectory planning using Mixed Reality: an experimental validation." 2021 20th International Conference on Advanced Robotics (ICAR). IEEE, 2021.

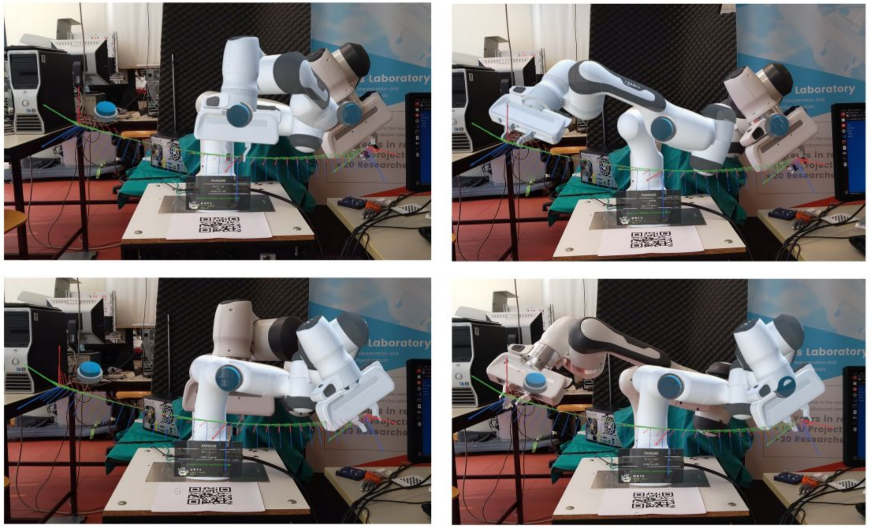


Fig. 5. Playback of the trajectory done by the virtual robot in the upper row, and the subsequent execution of the real robot in the lower row. The pictures were taken from the HoloLens 2 front camera.

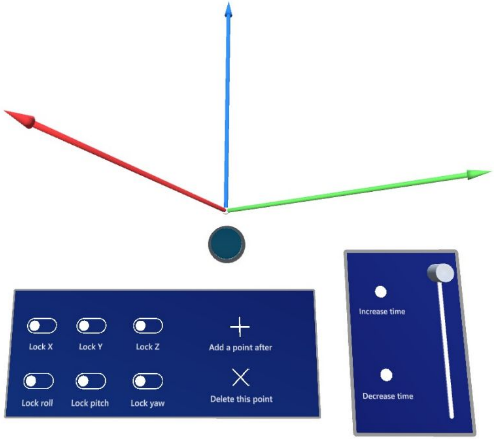


Fig. 6. Representation of a way point, with the frame defining the orientation, and the menus to interact with, in order to carefully design the trajectory.

In-depth analysis of related work

Let’s take a detailed look on other studies on path programming for robots:

Reference: Ostanin, Mikhail, et al. "Human-robot interaction for robotic manipulator programming in Mixed Reality." 2020 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2020.

System Description	Method	Evaluation
Hololens application connected to a UR and a KUKA robot, used for path generation, visualization and scaling. Seems like an extension of prior work from Ostanin. **Interesting feature:** uses the spatial mapping from Hololens to implement obstacle avoidance.	**1)** Implemented the prototype. **2)** Demonstrated the prototype features on a pick and place and contact operation tasks.	No evaluation was done, just a demonstration.

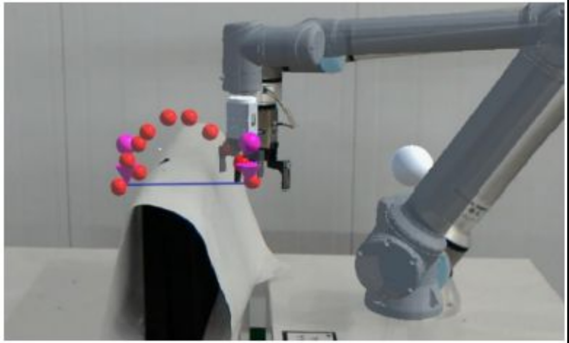
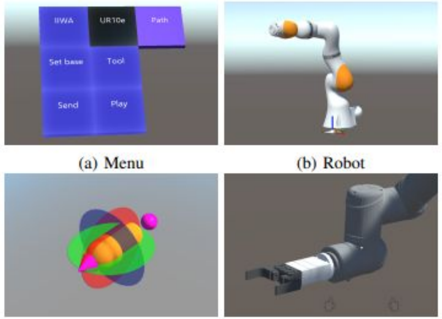


Fig. 5: Path with obstacle avoidance

In-depth analysis of related work

Let’s take a detailed look on other studies on path programming for robots:

Reference: Ostanin, Mikhail, et al. "Human-robot interaction for robotic manipulator programming in Mixed Reality." 2020 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2020.



(a) Menu (b) Robot
(c) Goal point (d) Gripper

Fig. 2: Main virtual object of the system

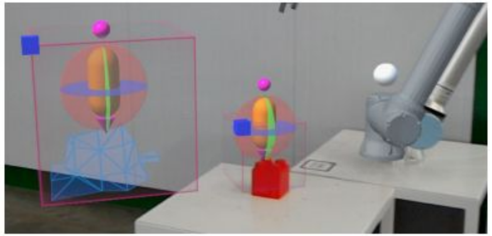


Fig. 6: Path scaling

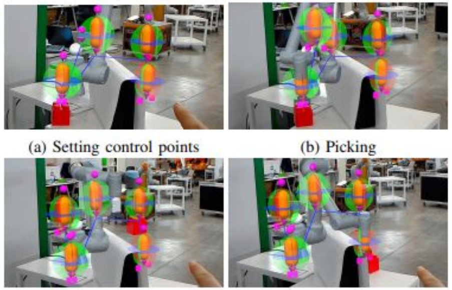

(a) Setting control points (b) Picking
(c) Execution (d) Placement

Fig. 7: The basics steps of draw words by Industrial robot programming using mixed reality

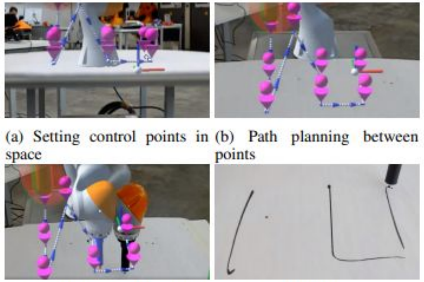

(a) Setting control points in space (b) Path planning between points
(c) Simulation and execution of drawing (d) Results

Fig. 8: The basics steps of draw words by Industrial robot programming using mixed reality

Overall analysis

Study	Uses Hololens?	Runs an experiment?	Is focused on end-users?
Ajaykumar, Gopika, Maia Stiber, and Chien-Ming Huang. "Designing user-centric programming aids for kinesthetic teaching of collaborative robots."	❌	✅	✅
Quintero, Camilo Perez, et al. "Robot programming through augmented trajectories in augmented reality."	✅	✅	❌
Ostanin, Mikhail, and Alexandr Klimchik. "Interactive robot programing using mixed reality."	✅	❌	❌
Piccinelli, Marco, et al. "Trajectory planning using Mixed Reality: an experimental validation."	✅	❌	❌
Ostanin, Mikhail, et al. "Human-robot interaction for robotic manipulator programming in Mixed Reality."	✅	❌	❌

What can we do different?

We can inspire our prototype on studies related to end-user programming (not necessarily related to robot programming), such as this one from Amy Ko et al:

> After classifying all of the different barriers that students encountered, there were six major barriers that accounted for our data:
>
> *Design* – Complex computational problems that users were not trained to solve, such as sorting and searching.
>
> *Selection* – Finding code, usually part of an API, that produces a desired behavior, such as tracking time.
>
> *Use* – Once some class, method, or data structure was found, learning how to properly use its programming interface, such as how to start and stop a timer.
>
> *Coordination* – Learning rules about how entities can communicate, such as how to send data between forms.
>
> *Understanding* – Forming hypotheses about the potential causes of a program's behavior.

Many studies on path programming for robotics out there are not made by researchers on Software Engineering. Many any of them did not use the knowledge existent in our field of study to build their methodologies. Can we use this in our favor?

Reference:
Ko, Amy J., Brad A. Myers, and Htet Htet Aung. "Six learning barriers in end-user programming systems." 2004 IEEE Symposium on Visual Languages-Human Centric Computing. IEEE, 2004.

What can we do different?

We can inspire our prototype on studies related to end-user programming:

The *13 Cognitive Dimensions of Notation (CDN)* [26] are a popular framework for analyzing visual languages. This framework describes how users understand and interact with with different visualizations. The CDN provides terminology to effectively describe and compare many different types of visual languages. We use its terminology in the following sections when describing our design decisions.[1]

Reference:
Ritschel, Nico, et al. "Comparing block-based programming models for two-armed robots." IEEE Transactions on Software Engineering (2020).