COM-480 – Milestone 2
Mixed reality toolkit for ROS visualization on Microsoft HoloLens

Tommaso Gargiani, Linyida Zhang, Zijun Cui

May 6, 2022

1 The Project

1.1 Types of Visualization

Data from different sensors is displayed in different ways. Note that some types of data are optional because the robot is not equipped with specific sensors. The possible visualization in our project could be classified into four categories:

1. Transform data. It is delivered by the /tf topic and describes the positions and poses of the robot and its components.
2. Point Cloud data. These kinds of data are usually collected through PointCloud or LaserScan, and demonstrate the surrounding information. They are drawn in local (or robot) frame.
3. Map data. Commonly used map data and costmap data are built through SLAM process, showing the cumulative perception of the robot. They are drawn in the world frame.
4. Others. Such as clock time and some debug logs.

1.2 Data Processing and Analysis

The focus of this project is really in the data visualization challenge regardless of the type of ROS topic the user wants to use.

A general framework for the project is shown in Figure 2. The most challenging, yet most interesting part of this project is the sensor visualization fusion to the real world. ROS sensors topics contain in their metadata information about the position of the sensors relative to an point in the robot or the world. Based on whether there is available transform data from the robot, the we can locate the visualization in two ways.

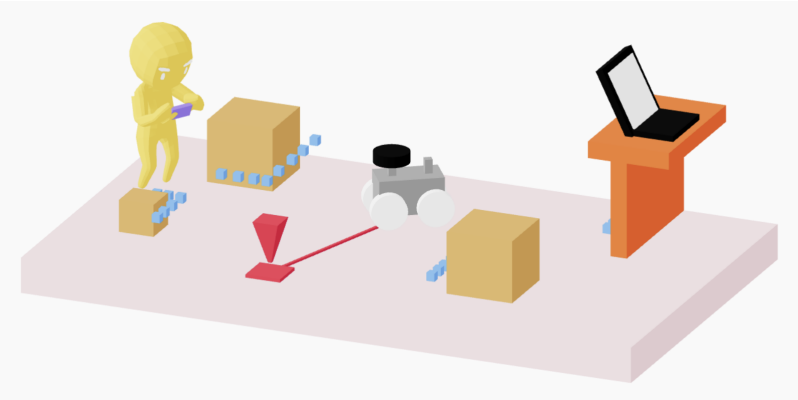


Figure 1: Sketch of a use-case of the app

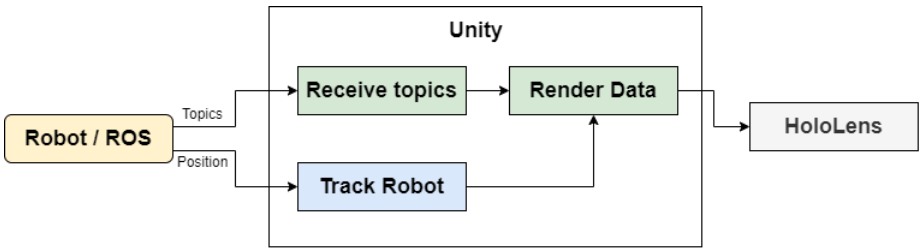


Figure 2: Pipeline for the visualization

1.2.1 World Anchor Method

In the easiest scenario, the running odometry estimators running on the robot provide usable transform data. As the robot runs, the position of the sensors in the world frame is known. The position and orientation for every visualization is handled automatically.

In this case the user can manually position and orient a static anchor, and place it to the one the ROS transforms are mapping to.

1.2.2 Robot Anchor Method

If the robot is not aware of its position, it is up to us to get an estimation of the robot odometry to map it in the augmented reality scene.

In this case, we track the robot position by exploiting undistorted images or 3D object tracking methods. The sensor visualizations are then transformed to the moving anchor that follows the robot.

1.3 User Interface

By default, the Unity Robotics Visualizations Package we use is controlled via a user interface called "Hud". The Hud is located in the top left corner of the screen and it usually has 3 buttons – "Topics", "Transforms" and "Layout". When clicked, each button displays a scrollable list of buttons that are used to select the desired attribute to visualize.

Unfortunately, since the Hud was specifically designed for Unity, we cannot use it to control the visualizations on HoloLens. Therefore, we have to create our own user interface using the Mixed Reality Toolkit, a library designed for HoloLens development.

On the other hand, creating a UI specifically for HoloLens is more convenient and will provide an improved user experience, since it is known that using static objects in the mixed reality (such as the Hud in the top left corner) frequently causes user discomfort and dizziness.

The new UI (see Figure 3) follows the same logic as the Hud – when the user selects the "Topics" button, a scrollable selectable list of available topics will appear under the form of buttons. Pressing a button will activate/deactivate the visualization of the given topic.

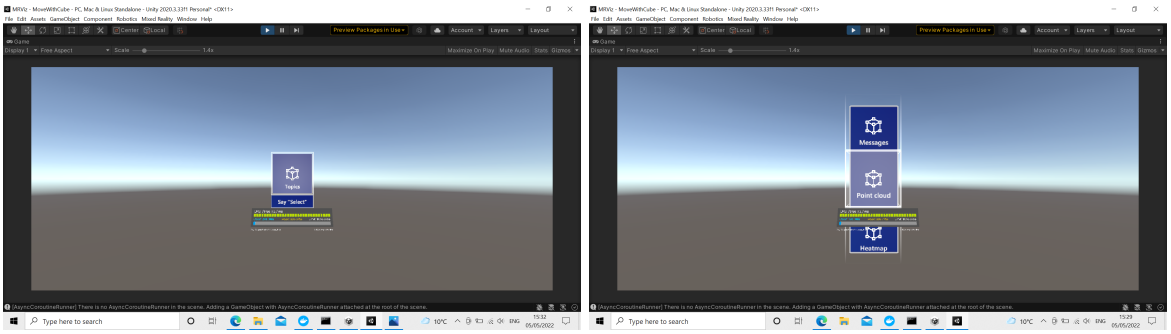
We chose not to include the other two buttons available in the Hud (Transforms and Layout), as they are not relevant for our project.

1.4 Webpage

The webpage will serve to present our data visualization results in different scenarios and to help visitors understand its working principle. The main subpages it will include are as below:

1. Demo page. The demo page will be the main page, where visitors can see the data visualizations in different scenarios. Videos exported from Unity will be uploaded here. By selecting different demos or topics, people can see the real-world scenario, the data visualization in Hololens and the ROS sensor topics separately or combined together, which will give them a visual understanding of our work.

2. Working principle page. This page will give a basic introduction of the tools we used and the aim we achieved, including explaining how to connect ROS and Unity, what the meanings of different topics in ROS sensors are, the two different ways to locate the visualization in the dimensions



(a) Topics button (b) Topic to visualize

Figure 3: HoloLens UI

of the real world, etc. Some 3D animations will be added as an auxiliary way to help people understand its working flow.

3. Set-up page. This page will give a detailed tutorial on how to visualize the different ROS topics with Hololens.

Click here to find our initial webpage.

2 Course correlations

In this project we develop visualization for a wide range of sensor information: heatmap/costmap, point clouds, odometry tracks, 2D laser scans, transform positions, etc. For each type of visualization we really highlight the mapping fusion of the data to the real world, which is the main topic of "Lecture 08 : Maps". We also keep in mind in our design process of all the tips and guidelines from other lectures, but our project being mostly in Augmented Reality we do not follow web data visualization development standards.

3 Tools

1. Communication between ROS and Unity: ROS TCP Connector Package.
2. ROS message visualization: Unity Robotics Visualizations Package.
3. Tracking the robot: Google Mediapipe toolbox or Vuforia SDK.
4. HoloLens configuration: Mixed Reality Toolkit

4 Milestones

- Render visualizations out of ROS/ROS2 topics. (Unity)
- Adjust the orientation and position of the sensors for a static world anchor and a dynamic one. (Unity)
- Get an estimation of the robot position in the real world for the dynamic anchor. (Instant tracking from Google Mediapipe toolbox)
- Develop the UI to select the desired topics to display and type/position/target of anchors. (MRTK package)
- Build a working demo with a compiled version with a laptop webcam.
- Test on real robots, instead of recordings. (Robot from Xplore)

- Make an interactive web-page to showcase the project, with different demos of the application (Next.js / React / three.js): https://mrviz.vercel.app/
- Build the app for Android. Optional. (Android SDK)
- Deploy the app on Hololens.

Bonus features:

- Track the robot using a more accurate method.
- Build the app for the browser. (Unity web assembly compiler)
- Allow multiple users to share the same view in the augmented reality scene. (Azure spacial anchor)