COM-480 – Milestone 1
Mixed reality toolkit for ROS visualization on Microsoft HoloLens

Tommaso Gargiani, Linyida Zhang, Zijun Cui

April 8, 2022

1 Problematic

In this project, we will create a mixed reality toolkit which will allow to visualize a robot's real-time perception of its environment, e.g. a LIDAR point cloud, in the physical world. To achieve this, we will use a Microsoft HoloLens headset. Finally, we will present the results of our work on a dedicated webpage.

Nowadays, engineers developing robotics software have a serious constraint. They can visualize the robot's perception of the environment, but cannot have a straightforward impression of the difference between such perceived data and the real physical world. They often have to rely on uninformative visualizations (Figure 1) of point clouds or heatmaps on a computer screen with no additional information about the robot's surroundings, making it rather difficult to find the flaws and debug the system.

Thanks to our toolkit, which overlays the robot's sensed data onto the real world, robotics engineers will benefit from seeing directly what their robot perceives by exploring the environment around them with HoloLens. Such feature will simplify the development of ROS (Robot Operating System) software, allowing engineers to fully leverage sensor data by visualizing them in the mixed reality.

We will do the project in Unity, a 3D game engine, which is officially supported to develop HoloLens applications.

2 Dataset

ROS provides standard services of an operating system, like hardware abstraction, commonly-used functionality, message-passing between processes etc. With ROS, these services are tailored to work with robots. It is also a framework of nodes, i.e. processes that perform computation, which allows each independent executable to be coupled at runtime. Nodes communicate with each other by publishing messages to named buses called topics. Almost all perceived data can be published and subscribed through different topics as messages. And such real-time message data flowing from the robot can also be recorded in bags, which will be our source of data.

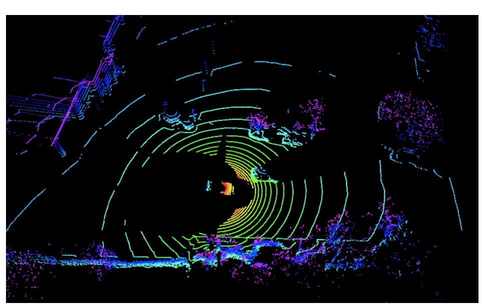
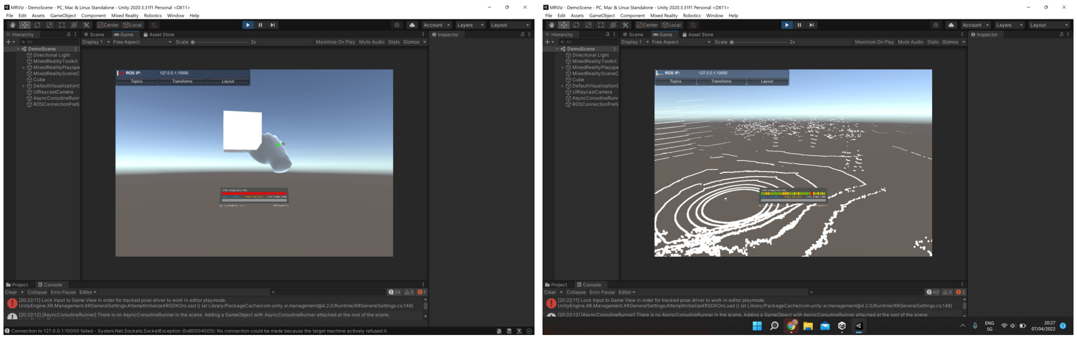


Figure 1: An example of a typical sensor data visualization.



(a) HoloLens integration (b) ROS integration

Figure 2: Current results

Thanks to the data logging capabilities of bags, we will be able to simulate a robot's functionality even without operating one and to playback key moments, which is a convenient and important feature during debugging. We will use the packages from Unity Robotics Hub to establish the connection between ROS and Unity, and then render the ROS messages we receive.

Throughout the semester, we will use robot simulators available in Unity in order to test our software. Later, we will borrow a robot from the EPFL Xplore project and a HoloLens headset from the EPFL CHILI laboratory. This will allow us to test the project more in depth and establish a working connection between our toolkit and HoloLens.

3 Data analysis

Since we are working with structured data, there is no need to do any pre-processing or analysis of the data. However, there are several types of messages we can potentially use – *odometry*, which estimate the robot's position and velocity in free space, *point cloud*, a collection of 3D points representing detected objects, *heatmap*, the magnitude of a phenomenon (the actual phenomenon depends on the employed sensor) visualized as color, etc.

We plan to visualize one type of messages for the time being, focusing on the other types later. Also, we are considering the possibility of combining together the visualizations of several message types, allowing more complex and informative evaluations of the robot's environment perception.

So far, we have tested the HoloLens integration and verified the connection between ROS and Unity separately (Figure 2).

Next, we will focus more on visualizing the perceived data, locating HoloLens, and tracking robot position.

4 Related work

Our project is inspired by RViz, a well-known data visualizer for ROS, and ROS#, which serves for communication to ROS from Unity.

As they lack the interaction between the ROS data and the real world, a better visualizer using mixed reality could be proposed.

Following the same idea, ARViz developed the toolchain in Unity and used HoloLens to visualize the robot's environment. However, it cannot track the position of robot. Our current results have the similar issue (Figure 2): The shown data is drawn based on the local coordinate frame (or the robot coordinate frame), which is always located in (0, 0, 0) of the world coordinate frame and doesn't change as the robot moves. So we also plan to implement the position tracking function to make the data move along with the robot in the mixed reality.

As for the mixed reality part, the Mixed Reality Toolkit is a project designed for HoloLens app development, using which it is possible to fully reach the potential of HoloLens's capabilities.

Therefore, we will start our work on top of several related projects, building it as a combination of the mixed reality visualization and robot tracking packages.