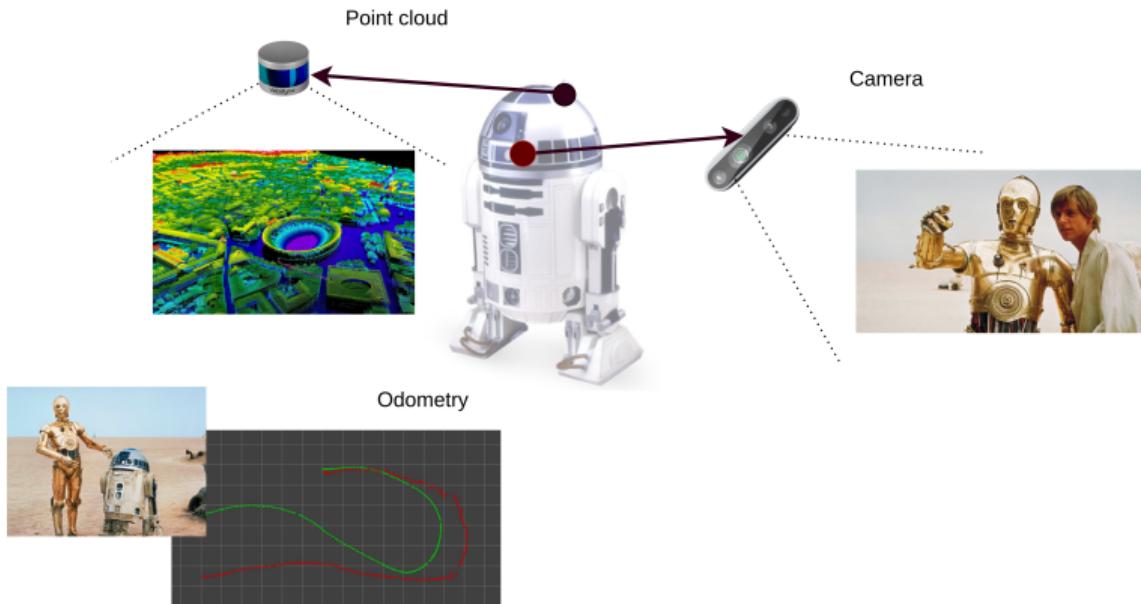


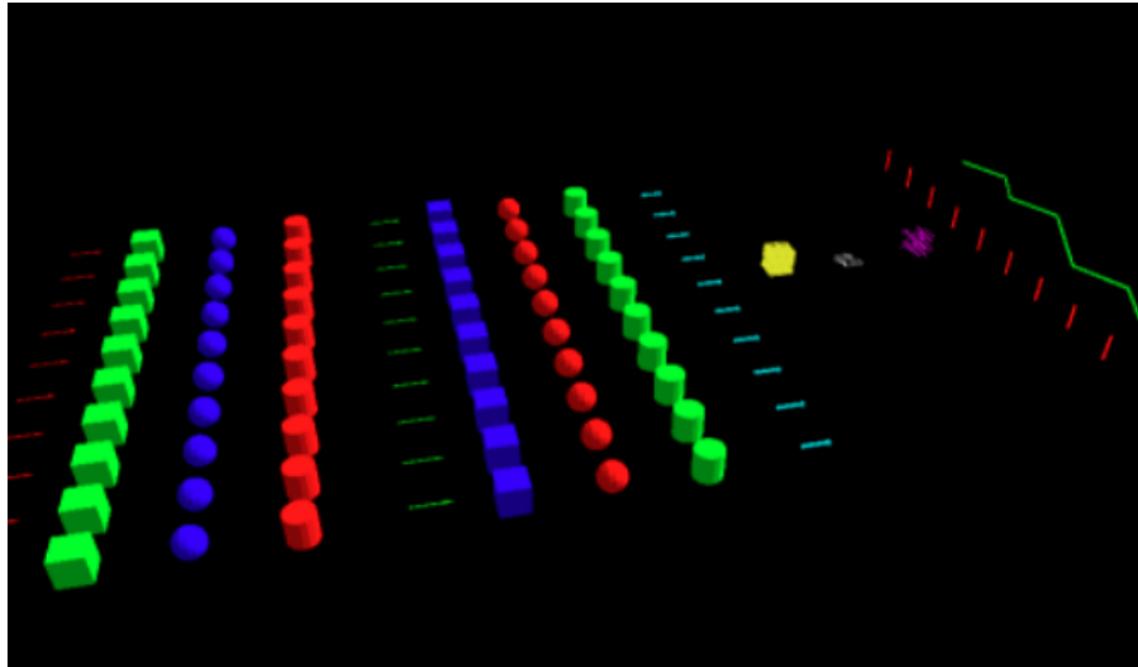
Introduction to ROS2: Basics, Motion, and Vision

ROS Visualization

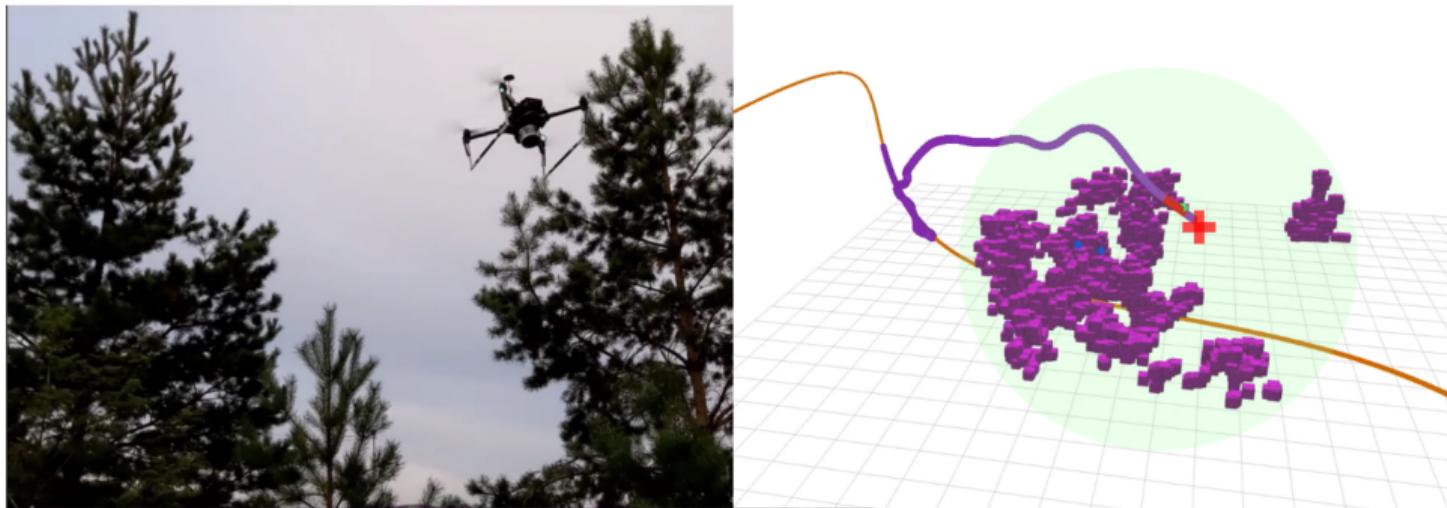


ROS Visualization

1 rviz <https://github.com/ros2/rviz>



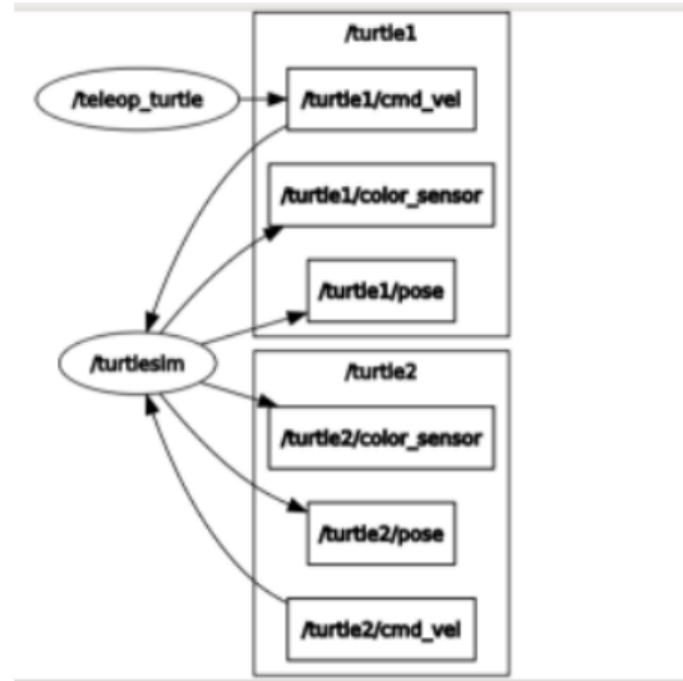
ROS Visualization



- Initial reference trajectory R
- Refined reference trajectory R_{t_n}
- Prediction horizon P_{t_n}
- Map update range (M)
- Current pose

ROS Visualization

1 rqt <https://docs.ros.org/en/foxy/Concepts/About-RQt.html>



ROS Visualization with rqt

- 1 rqt can be used as a visualization tool as well as debugging tool
- 2 Scalar data can be easily visualized with rqt, however, non-scalar data such as images, pose, can also be visualized up to some extent in default setting
- 3 rqt consists of several sub-tools : **rqt**, **rqt_bag**, **rqt_console**, **rqt_dep**, **rqt_graph**, **rqt_image_view**, **rqt_logger_level**, **rqt_plot** and **rqt_shell**

ROS Visualization with rqt

- 1 Let's try to plot scalar data

to plot scalar data using rqt

```
ros2 launch ros_visualization joy_hagen_sub.launch.py  
ros2 launch ros_visualization joy_hagen_pub.launch.py
```

- 2 If the message consists of data that is not a scalar value, have to specify the fields separated by a comma,

to plot non-scalar data

```
/joy_hagen imu/x
```

ROS Visualization with rqt

to visualize a single image

```
ros2 launch ros_visualization hagen_joy_camera.launch.py
```

ROS Visualization with rviz

- 1 3D visualization is not possible with rqt, rviz, which is built with the help of OpenGL, can be used to visualize the 3D sensor data, e.g., stereo images, point clouds from camera
- 2 rviz can be launched in different ways:

to start rviz

rviz2

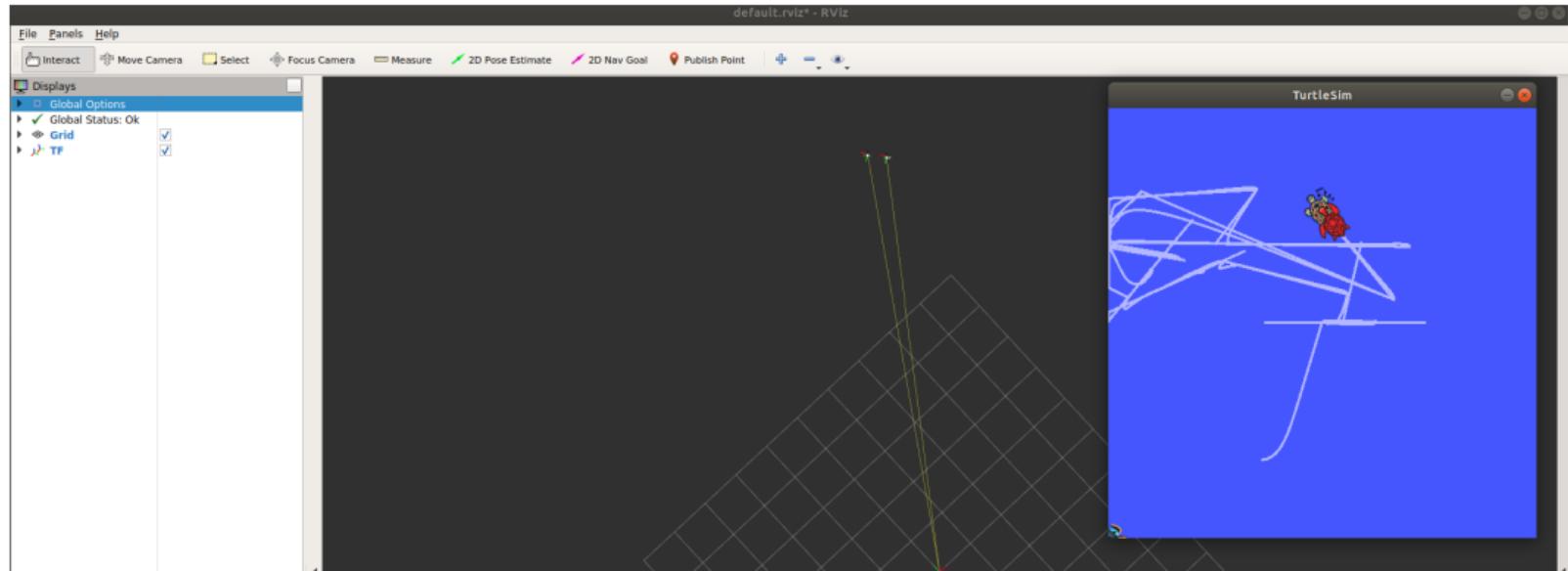
ROS Visualization: topics and frames

- 1 All the topics must have a frame_id, which defines the location of the sensor or device with respect to defined coordinate system
- 2 Usually, when we define the root of robot, it is named as the **base_link**, which is the common standard that everyone follows

to visualize frame of robots

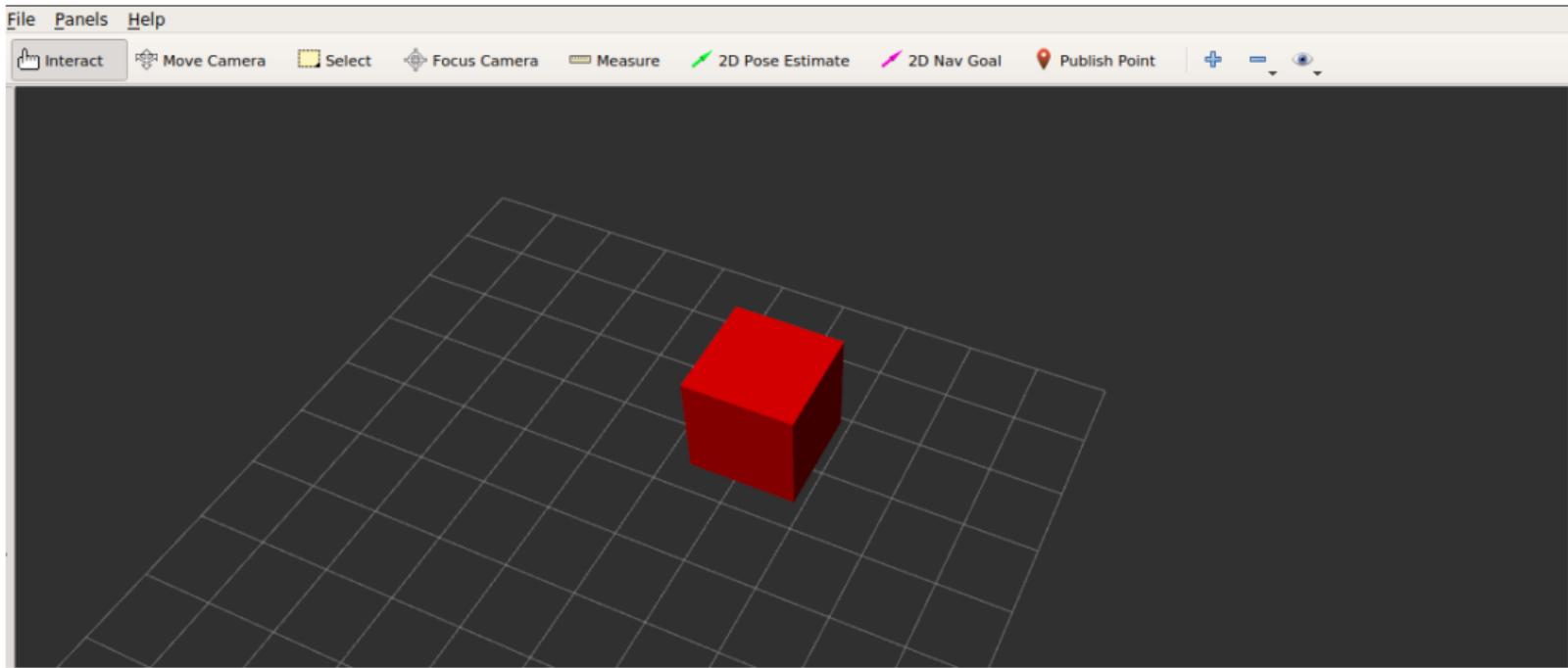
```
sudo apt-get install ros-humble-turtle-tf2-py ros-humble-tf2-tools ros-humble-tf-  
transformations  
ros2 run tf2_tools view_frames  
ros2 run tf2_ros tf2_echo turtle2 turtle1  
ros2 run rviz2 rviz2 -d $(ros2 pkg prefix --share turtle_tf2_py)/rviz/turtle_rviz.rviz
```

ROS Visualization: topics and frames

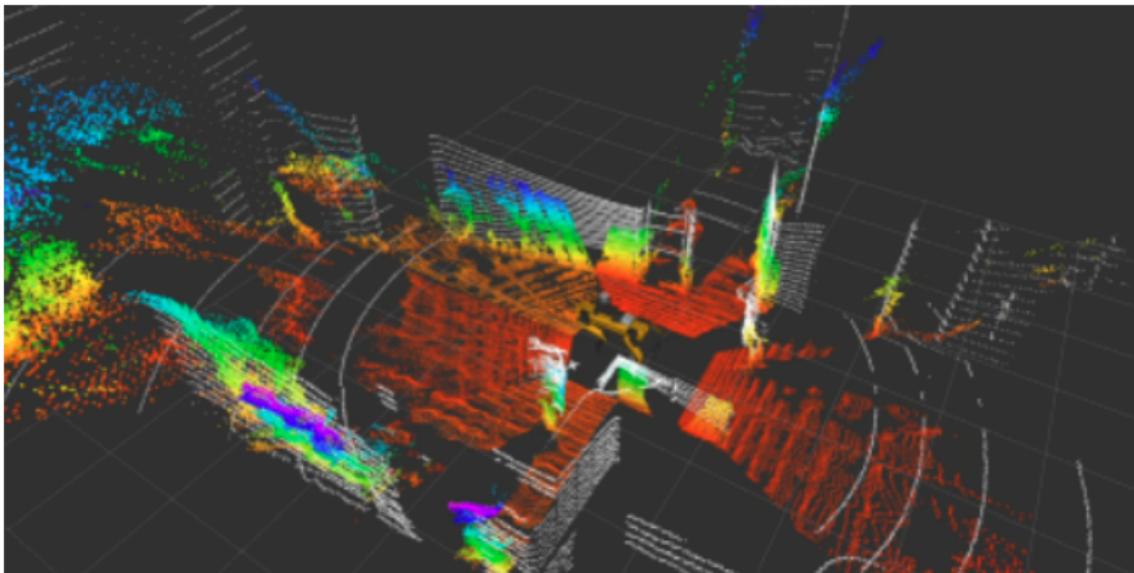


ROS Visualization: cube

Let's try to visualize a cube <http://wiki.ros.org/rviz/DisplayTypes/Marker>



ROS Visualization: point cloud



ROS Visualization: point cloud

Let's try to visualize a point cloud

