**Autoware Installation**

In this project, we have installed Autoware using the procedures shown below and achieve mapping, localization, and path following function using instructions below. To refer for more information about the installation process, please visit the Autoware official documentation page.

<https://gitlab.com/autowarefoundation/autoware.ai/autoware/-/wikis/Source-Build>

A screenshot of a cell phone

Description automatically generated

A screenshot of a cell phone

Description automatically generated

Since our system is Ubuntu 18.04 and our ROS version is Melodic, we installed Autoware v1.12.0 using command below.

$ sudo apt update

$ sudo apt install -y python-catkin-pkg python-rosdep ros-$ROS\_DISTRO-catkin

$ sudo apt install -y python3-pip python3-colcon-common-extensions python3-setuptools python3-vcstool

$ pip3 install -U setuptools

For installation instructors for CUDA 10.0, see <https://docs.nvidia.com/cuda/archive/10.0/cuda-installation-guide-linux/index.html>

1. Create a workspace

$ mkdir -p autoware.ai/src

$ cd autoware.ai

2. Download the workspace configuration for Autoware AI

For the 1.12.0 release:

$ wget -O autoware.ai.repos "https://gitlab.com/autowarefoundation/autoware.ai/autoware/raw/1.12.0/autoware.ai.repos?inline=false"

3. Download Autoware.AI into the workspace.  
$ vcs import src < autoware.ai.repos

4. Install dependencies using rosdep

$ rosdep update

$ rosdep install -y --from-paths src --ignore-src --rosdistro $ROS\_DISTRO

5. Compile the workspace (with CUDA support)

$ AUTOWARE\_COMPILE\_WITH\_CUDA=1 colcon build --cmake-args -DCMAKE\_BUILD\_TYPE=Release

Note: In our installation, we have faced several problems of conflicting library, we resolve it by removing anaconda from our Ubuntu system and it resolved any conflicting library.

**Autoware Runtime Manager Instruction**

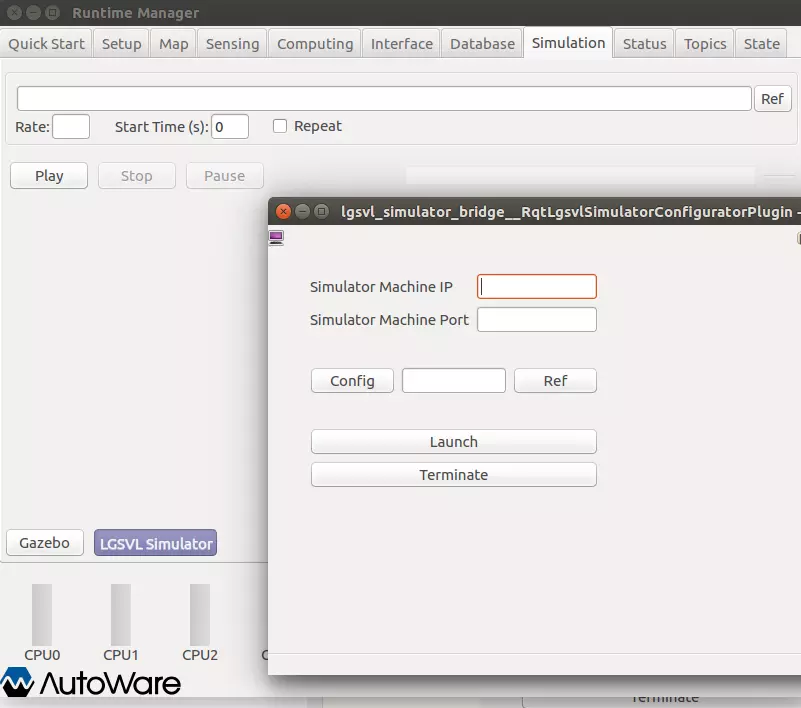
To start the autoware runtime manager, type in below function

$ cd autoware.ai

$ source install/setup.bash

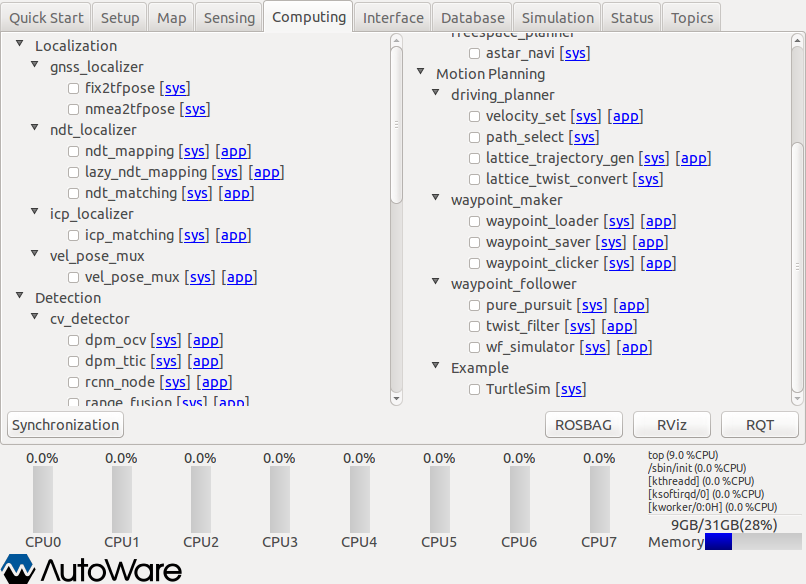
$ roslaunch runtime\_manager runtime\_manager.launch

Under the Simulation Tab in the runtime manager, select the simulator button



Enter the IP address and port number of the computer which the LGSVL simulator is running on and click launch.

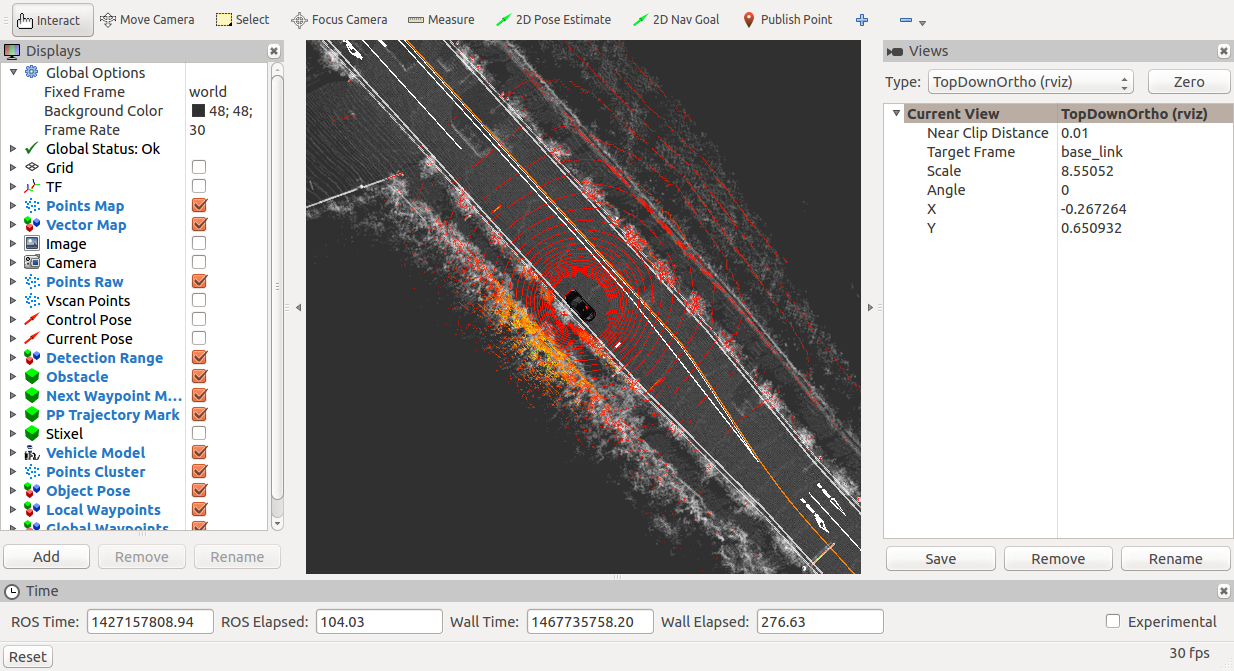
To perform mapping, turn on the “ndt\_mapping” under the computing tab and the LIDAR sensor in the LGSVL simulator. The point cloud map will be saved in designated path as a .pcd file.



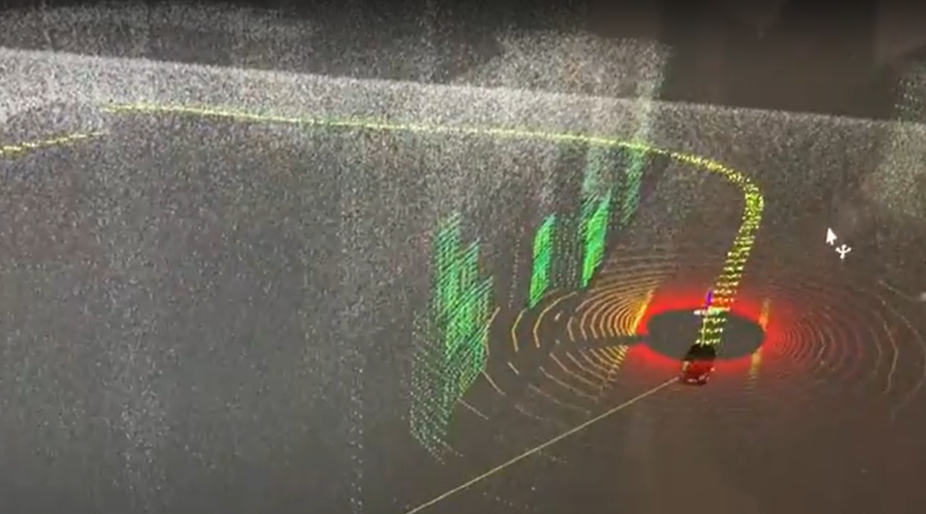
A person sitting at a desk in front of a computer

Description automatically generated

For Localization, first load the point cloud map under the map tab. Then, turn on the “ndt\_maching” under the computing tab and the LIDAR sensor in the simulator. The vehicle might not start at the right position. Therefore, open the RVIZ window, select the “2D Pose Estimate” and click on the current position of the car to initialize the localization process.



For the pure pursuit path planning, first a csv path file is required. Select the “waypoint\_saver” under the computing tab and click on the word “app” to select save path. Next, start driving the vehicle in the simulator and the driving path will be saved in a csv file in selected save directory.



To start the pure pursuit, load the map and enable the localization function. Next, turn on “voxel\_grid\_filter” in the sensing tab to increase the localization performance. Then, turn on “vel\_pose\_connect”, “waypoint\_loader” with previously saved path file, “lane\_rule”, “lane select”, “lattice\_velocity\_set”, “path\_select”, “pure\_pursuit”, and driving especially turning. Finally, the vehicle in the simulator should start moving according to the previous path without any problem.

A close up of a computer

Description automatically generated

For detection, we are using yolov3 to detect cars, pedestrian, and traffic light. However, in order to so, a yolov3 weights file is required to perform the object detection. In our originally built code, there is no yolov3 weights file included. Therefore, the yolov3 weights and config file from below online resource are used.

<https://github.com/lgsvl/autoware-perception/tree/master/yolov3>

Of course, the main camera in the simulator must be turned on. In the RVIZ window tabs, click on tools->add new panel. In the add new panel window, select the “ImageViewerPlugin” and a new image panel should be added to your RVIZ window. Select the image topic and the latest image with object detection bounding box will be shown in the RVIZ image panel.

A screenshot of a computer

Description automatically generated