

# 轮 趣 科 技

## Autoware 自动驾驶 使用教程

推荐关注我们的公众号获取更新资料



版本说明:

版本	日期	内容说明
V1.0	2022/08/30	第一次发布
V1.1	2023/06/26	增加 GNSS 辅助定位内容
V1.2	2023/11/16	增加 docker 安装版本, 重点参数标粗
V1.3	2024/07/09	信息勘误, GNSS 定位内容摘选成单独文档

网址:[www.wheeltec.net](http://www.wheeltec.net)

## 序言

本文档主要讲解了自动驾驶软件 Autoware 的基本使用方法。文档主要分为三个部分：第一部分主要是对 Autoware 的简单介绍；第二部分主要讲述了 Autoware 的具体应用，包括了建图、地图定位、雷达目标识别、open\_planner 导航、路径点导航、寻迹等；第三部分主要简单介绍高精地图的绘制。

# 目录

序言 .....	2
1. Autoware 介绍 .....	4
1.1 Autoware 简介 .....	4
1.2 Autoware 各模块节点功能简介 .....	5
1.3 Autoware 重要话题介绍 .....	8
2. Autoware 基础功能使用 .....	8
2.0 docker 安装方式与源码安装方式使用区别 .....	8
2.1 录制 ROSBAG .....	11
2.2 使用 ROSBAG 进行建图 .....	14
2.3 使用 ROSBAG 生成路径点 .....	21
2.4 地图定位 .....	25
2.5 雷达目标识别 .....	31
2.6 open_planner 导航 .....	33
2.7 路径点导航 .....	39
2.8 寻迹 .....	40
3. MapTools 工具绘制高精地图 .....	45
3.1 环境适配 .....	45
3.2 高精地图简单绘制 .....	47

# 1. Autoware 介绍

## 1.1 Autoware 简介

Autoware 是世界上第一款开源的自动驾驶框架。Autoware 最早是由名古屋大学研究小组在加藤伸平教授的领导下于 2015 年 8 月正式发布。Autoware 提供了一套丰富的自动驾驶模块，主要功能包括了定位、映射、目标检测和跟踪、任务和运动规划、车辆控制等。本文基于构建在 ROS 平台上的 Autoware.AI 的 1.13 版本实现，主要架构如下图所示。

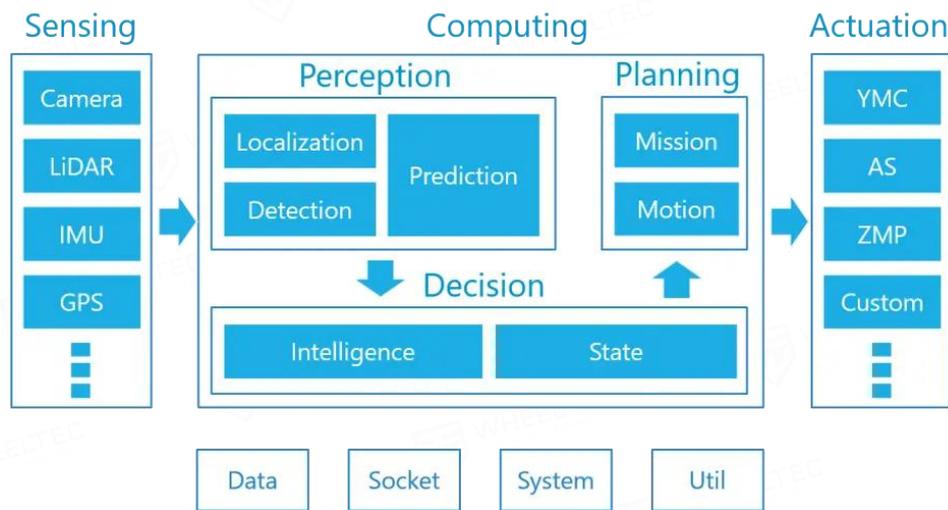


图 1-1-1 Autoware 框架

Autoware 主要包含 sensing、computing (preception、planning、decision)、actuation 几个部分。

(1) Sensing 模块主要是利用各种传感器采集真实世界的各类数据，如 camera 采集图像、LiDAR 采集激光点云数据、IMU 采集小车的当前位姿等等，采集到的数据需要传输给 computing 模块进行处理。

(2) Computing 模块主要是对传感器的数据进行加工处理，以达到安全导航的目的，该模块也可以分为 3 个小模块：preception (感知模块)、decision (决策模块)、planning (规划模块)。

A. Preception (感知模块)：首先基于 3D 点云地图，利用雷达等传感器采集到的数据通过 localization 模块进行定位，然后 detection 模块主要检测当前场景下存在的障碍物，如车辆、行人等，然后 predition 模块对检测到的

障碍物预测未来运动，以达到提前避让的效果。

B. **decision**（决策模块）：根据感知模块感知的结果，Autoware 决策一个由有限状态机表示的驾驶行为，可以根据当前状态选择合适的计划功能。

C. **planning**（规划模块）：主要根据当前决策的起始点和目标点，采用 **mission** 模块和 **motion** 模块计算出一条路径。

(3) **Actuation** 模块表示驱动器模块，主要接受 **planning** 模块计算出的结果，然后经过驱动器控制小车运动。

## 1.2 Autoware 各模块节点功能简介

### ① Localization（自定位）

#### A. **lidar\_localizer**

**lidar\_localizer** 使用 LIDAR 的扫描数据和预先构建的地图信息，计算车辆当在全局坐标的当前位置(x,y,z,roll,pitch,yaw)。autoware 使用正态分布变换(NDT)算法来匹配激光雷达当前帧和 3D map。

#### B. **gnss\_localizer**

**gnss\_localizer** 将 NMEA 消息从 GNSS 接收器转换到(x,y,z,roll,pitch,yaw)位置。该结果可以单独用作车辆的位置，也可以用于车辆的初始化位置和对 **lidar\_localizer** 车辆位置的补充修正。

#### C. **dead\_reckoner**

**dead\_reckoner** 主要使用 IMU 传感器预测车辆的下一帧位置，也可以用来对 **lidar\_localizer** 和 **gnss\_localizer** 的结果进行插值。

### ② Detection（检测）

#### A. **lidar\_detector**

从激光雷达单帧扫描读取点云信息，提供基于激光雷达的目标检测。主要使用欧几里德聚类算法，从地面以上的点云得到聚类结果。除此之外，可以使用基于卷积神经网络的算法进行分类。

#### B. **vision\_detector**

读取来自摄像头的图片，提供基于图像的目标检测。主要的算法包括 R-CNN，SSD 和 YOLO，可以进行多类别(汽车，行人等)实时目标检测。

### C. vision\_tracker

对 vision\_detector 的检测结果完成目标跟踪功能。算法基于 Beyond Pixels，图像上的目标跟踪结果被投影到 3D 空间，结合 lidar\_detector 的检测检测结果输出最终的目标跟踪结果。

### D. fusion\_detector

输入激光雷达的单帧扫描点云或者摄像头的图片信息，在 3D 空间中实现更准确的目标检测。激光雷达的位置和摄像头的位置需要提前进行联合标定，现在主要是基于 MV3D 算法来实现。

### E. fusion\_tools

将 lidar\_detector 和 vision\_detector 的检测结果进行融合，将 vision\_detector 的识别类别信息添加到 lidar\_detector 的聚类结果上。

### F. lidar\_tracker

预测检测目标的下一步位置，跟踪的结果可以被进一步用于目标行为分析和目标速度分析。跟踪算法主要是基于卡尔曼滤波器。

## ③Prediction (预测)

### A. naive\_motion\_predict

使用 lidar\_tracker 的结果来进一步预测物体的运动方向和预测的道路。

## ④Decision (决策)

### A. Decision\_maker

Decision\_maker 订阅了大量的感知结果、地图信息和当前状态有关的话题，以便发布下一时刻的状态，状态更改将影响 planning 规划结果

### B. state\_machine

state\_machine 预先定义的规则范围内改变状态，编排 decision\_maker。

## ⑤Mission Planning (任务规划)

### A. route\_planner

用于寻找从起始点到目标点的全局路径，该路径由道路网中的一系列十字路口组成。

### B. lane\_planner

根据 `route_planner` 发布的一系列十字路口结果，确定全局路径是由哪些 lane 组成，lane 是由一系列的 `waypoint` 点组成。

#### C. `waypoint_planner`

用于产生到达目标点的一系列 `waypoint` 点，它与 `lane_planner` 不同的地方在于，它发布的是单一到达目标点的 `waypoint` 路径，而 `lane_planner` 是发布到达目标点的一系列 `waypoint` 数组。

#### D. `waypoint_maker`

是一个用于保存和加载手动制作的 `waypoint` 路径点文件的工具。车辆行驶过程中的路径分解成一系列的定位点信息和速度信息，储存汇总成为一个路径文件，加载该路径文件可以发布路径点信息给规划模块，使车辆沿着该路径行驶。

### ⑥Motion Planning (运动规划)

#### A. `velocity_planner`

更新车辆速度信息，注意到给定跟踪的 `waypoint` 里面是带有速度信息的，这个模块就是根据车辆的实际状态进一步修正速度信息，以便于实现在停止线前面停止下来或者加减速等等。

#### B. `astar_planner`

实现 Hybrid-State A 查找算法，生成从现在位置到指定位置的可行轨迹，这个模块可以实现避障，或者在给定 `waypoint` 下的急转弯，也包括在自由空间内的自动停车。

#### C. `adas_lattice_planner`

实现了 State Lattice 规划算法，基于样条曲线，事先定义好的参数列表和语义地图信息，在当前位置前方产生了多条可行路径，可以被用来进行障碍物避障或车道线换道。

#### D. `waypoint_follower`

这个模块实现了 Pure Pursuit 算法来实现轨迹跟踪，可以产生一系列的控制指令来移动车辆，这个模块发出的控制消息可以被车辆控制模块订阅，或者被线控接口订阅，最终就可以实现车辆自动控制。

## 1.3 Autoware 重要话题介绍

### ① 订阅雷达话题: /points\_raw

Autoware 默认使用 velodyne 雷达, 默认订阅的雷达话题为 /points\_raw, 雷达话题的 frame\_id 为 velodyne。使用雷达驱动包时, 雷达话题名称与 frame\_id 必须与上述的一致。

### ② 发布小车速度话题: /twist\_cmd

Autoware 通过 /twist\_cmd 发布小车的线速度与角速度信息来控制小车的运动, 该话题的消息格式为 geometry\_msgs::TwistStamped。

### ③ 订阅小里程计话题: /vehicle/odom

Autoware 中 Localization 自定位模块使用正态变换算法来匹配雷达当前帧和 3D 地图来实现定位, 同时也可以勾选使用里程计来辅助定位, 默认订阅里程计话题为 /vehicle/odom 话题, 使用到里程计定位时, 需发布话题为 /vehicle/odom 的里程计话题。

## 2. Autoware 基础功能使用

Autoware 在 home 目录下面的 autoware.ai 文件夹当中, 我们所有 autoware 导航相关的功能包都是在这个文件夹里面。建议初学者参数设置与教程截图一致, 然后按照教程步骤打开相关节点即可。

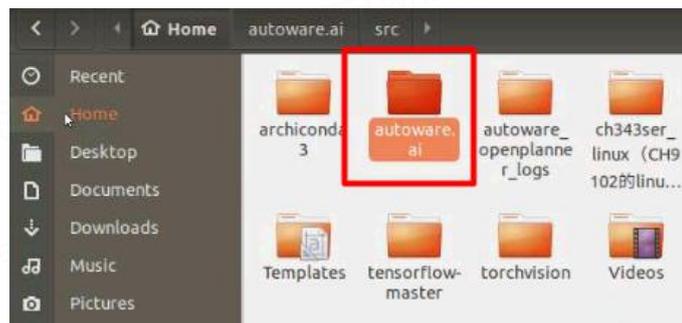


图 2-0-1 Autoware 文件夹

## 2.0 docker 安装方式与源码安装方式使用区别

Autoware 安装方式有两种。一种是在自己系统上安装 autoware 环境, 然后对 autoware 源码进行编译得到; 另一种是使用官网已经弄好 autoware 相关环境的

docker 镜像，我们只需要下载使用这个 docker 镜像即可。

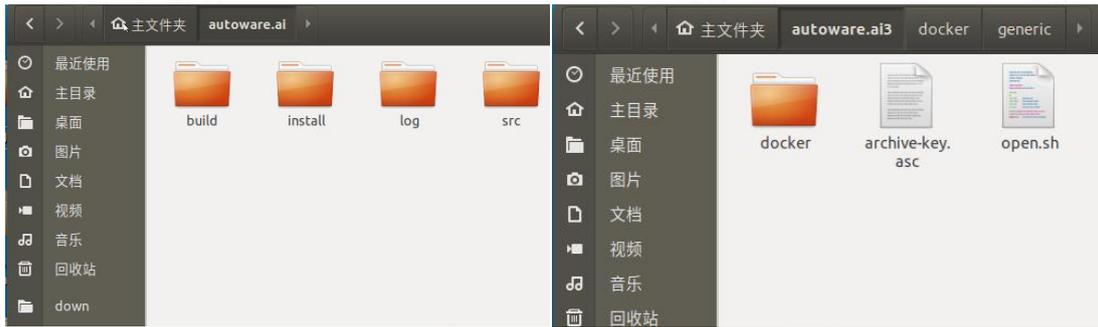


图 2-0-1 左：源码安装方式 右：docker 安装方式

我们主控中，Jetson Xavier NX 采用源码安装方式，Jetson Orin NX 采用的是 docker 安装方式。

两种安装方式在使用方面有一定的区别。

- 运行方式差别。使用源码安装方式，直接在小车系统命令窗口中运行 `roslaunch runtime_manager runtime_manager.launch` 即可打开 autoware 界面。使用 docker 安装方式，必须先在 autoware 目录下运行 `sudo ./open.sh` 进入 docker 容器中，然后在 docker 镜像环境中运行 `roslaunch runtime_manager runtime_manager.launch` 才能打开 autoware 界面。

使用 docker 安装方式时，需要运行 `exit` 退出 docker 容器然后才可以关闭命令行，避免产生过多无用容器消耗系统内存（可以运行 `sudo docker container prune` 手动清理无用容器）

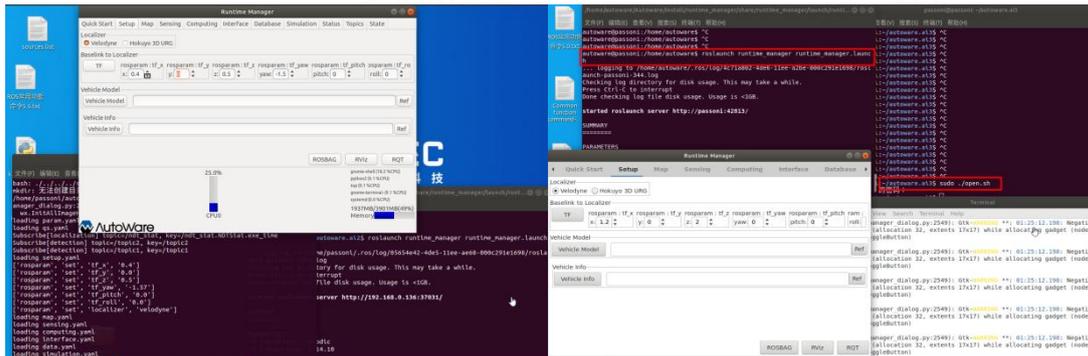


图 2-0-2 左：源码安装方式运行 右：docker 安装方式运行

- 保存文件差别。需要注意的是，Docker 镜像在退出时，仅能保存 `~/Autoware` 及 `~/autoware_shared_dir` 两个文件夹下的所有东西（包括对于 autoware 代码修改编译，autoware 界面相关参数的修改等），其余会恢复到初始状态，在此期间做的其他修改都会被清除掉，docker 安装方式的若保存 bag 包、

地图、路径点等文件建议保存在 `autoware_shared_dir` 目录下面，此目录与我们小车系统中的 `/home/wheeltec/autoware_shared_dir` 目录是共享的。

**Docker 安装方式**若想保存 docker 系统上的所有修改,可按照以下步骤操作。

- 进入 docker 镜像中进行一系列修改后，暂时不用运行 `exit` 退出 docker 容器
- 在小车系统命令行中运行 `sudo docker ps -a` 查看最新创建的容器，获取该容器的 id

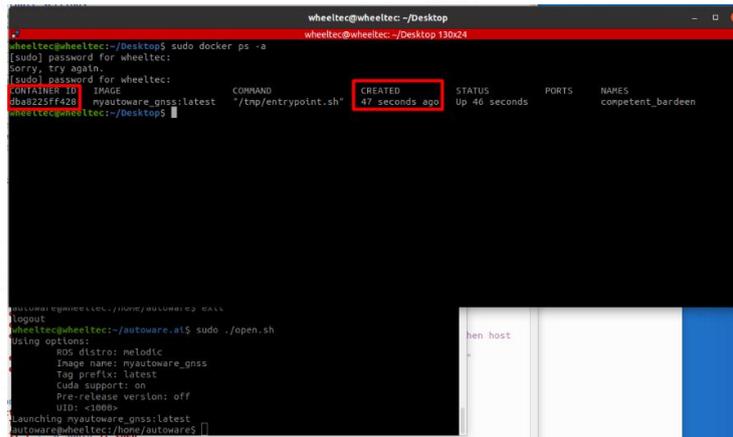


图 2-0-3 查看所有容器

- 按照以下格式修改运行，把修改后的容器保存成新 docker 镜像[`sudo docker commit 容器 id 新镜像名字:标签`]，新镜像名字和标签可以根据自己需求修改，如原 docker 镜像名字为 `myautoware_gnss`，标签为 `latest`
- 修改 `~/autoware.ai/docker/generic/run.sh` 文件中的 `IMAGE_NAME` 及 `TAG_PREFIX` 参数分为新镜像名字和标签（可运行 `sudo docker images` 获取当前所有镜像）。最后在之前打开的 docker 命令行中运行 `exit` 退出然后重新打开即可。



图 2-0-4 修改 run.sh 文件

- 打开 **rviz** 差别。源码安装方式在 **autoware** 界面中点击 **rviz** 即可打开 **rviz**，但在 **docker** 中的 **autoware** 界面上打开 **rviz** 会有问题，需要在小车系统中运行 **rviz**，然后点击左上角 **File**——**open config**，选择 `/home/wheeltec/autoware_shared_dir/default.rviz`

## 2.1 录制 ROSBAG

录制 **ROSBAG** 主要是为了后面的建图以及生成导航的路径点文件需要。  
**Autoware** 建图方式是利用 **roslaunch** 包进行离线建图(建议用户 **vnc** 远程小车桌面操作)。

### ①运行以下命令启动雷达、STM32、Autoware:

- 源码安装方式:

启动雷达、STM32、Autoware:

```
roslaunch turn_on_wheeltec_robot open_autoware.launch
```

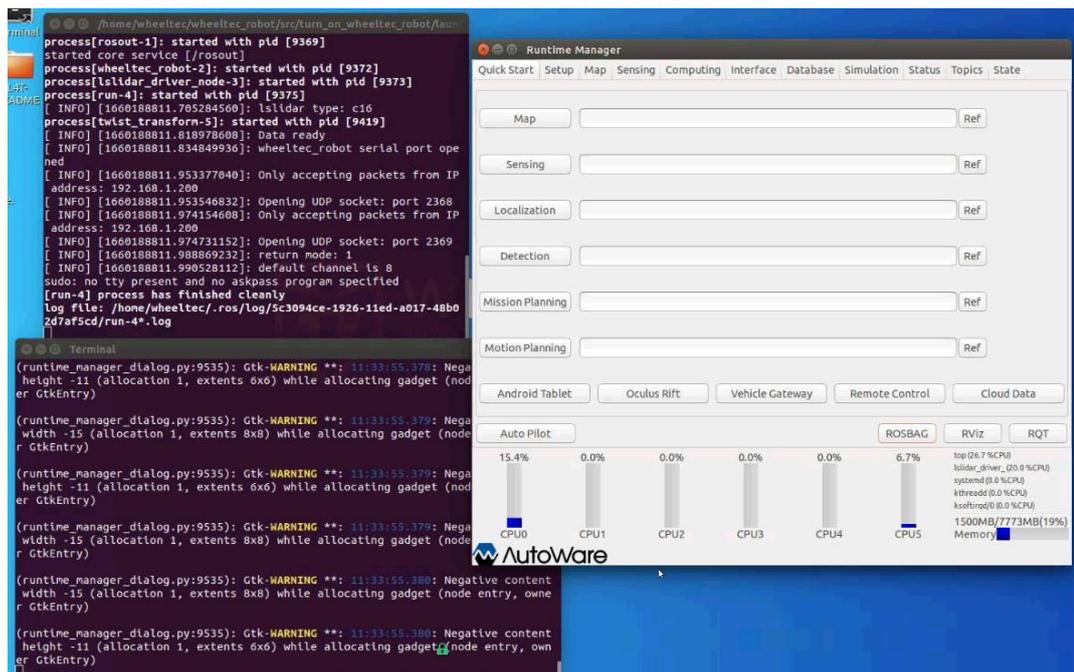


图 2-1-1 Autoware 启动页面

- Docker 安装方式:

启动雷达、STM32:

```
roslaunch turn_on_wheeltec_robot open_autoware.launch
```

启动 **autoware**:

在 `/home/wheeltec/autoware` 目录下运行 `sudo ./open.sh` 打开 **docker** 后运行

roslaunch runtime\_manager runtime\_manager.launch

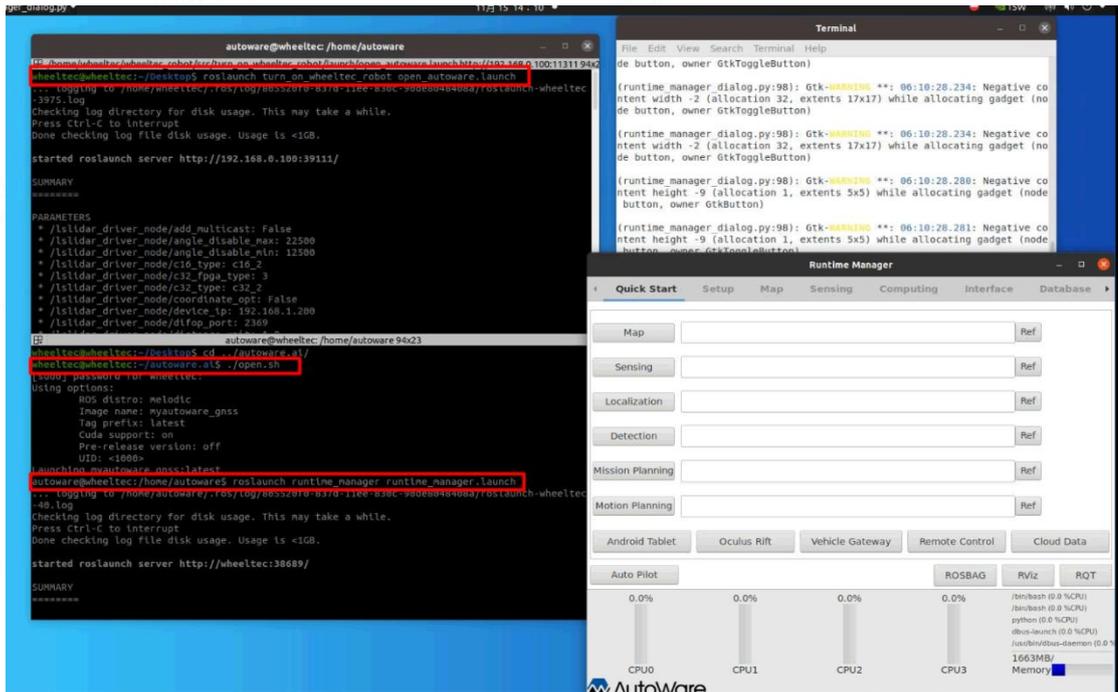


图 2-1-2 Autaware 启动页面

## ②录制 rosbag:

在 Autaware 界面中，点击[ROSBAG]按钮，在弹出来的界面中点击[Ref]选择 ROSBAG 的存储路径（docker 安装方式的注意保存在 autoware\_shared\_dir 目录下），然后勾选 imu 话题/imu\_raw、雷达话题/points\_raw、里程计话题 /vehicle/odom，然后点击“start”即可开始录制。

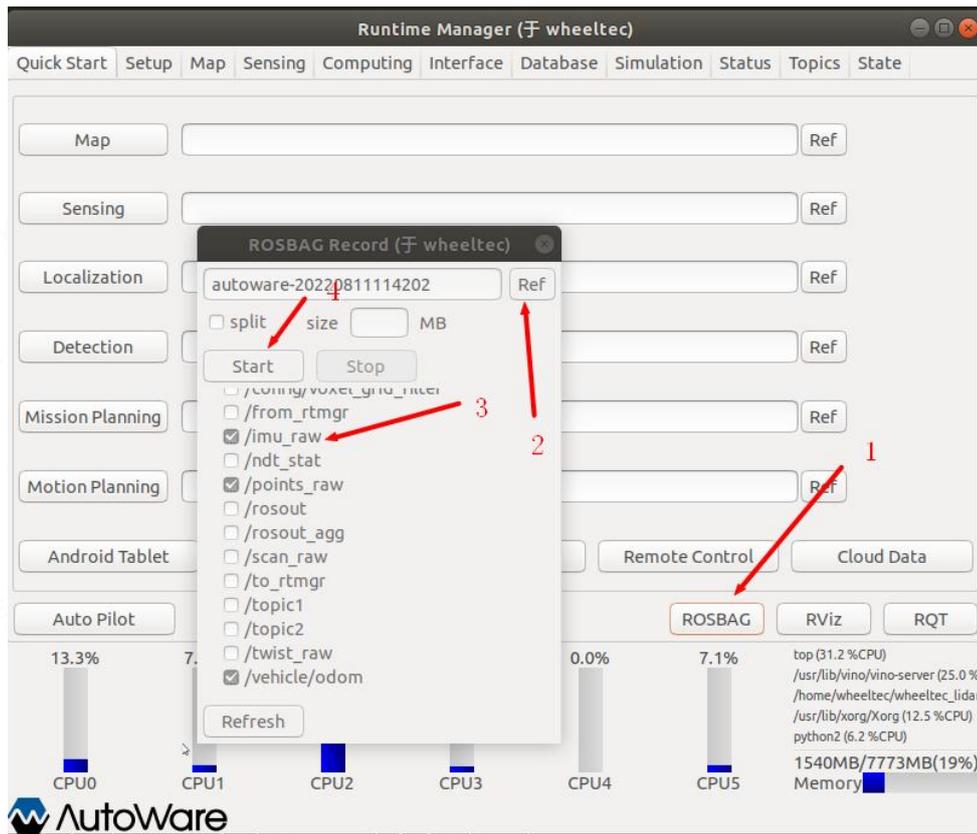


图 2-1-3 Autoware 录制 ROSBAG

### ③控制小车运动:

开始录制后, 就可以控制小车运动了, 可以通过运行键盘控制命令控制小车运动:

```
roslaunch wheeltec_robot_rc keyboard_teleop.launch
```

录制完成后点击[stop]按钮便可结束录制, 录制的 bag 文件会自动保存。

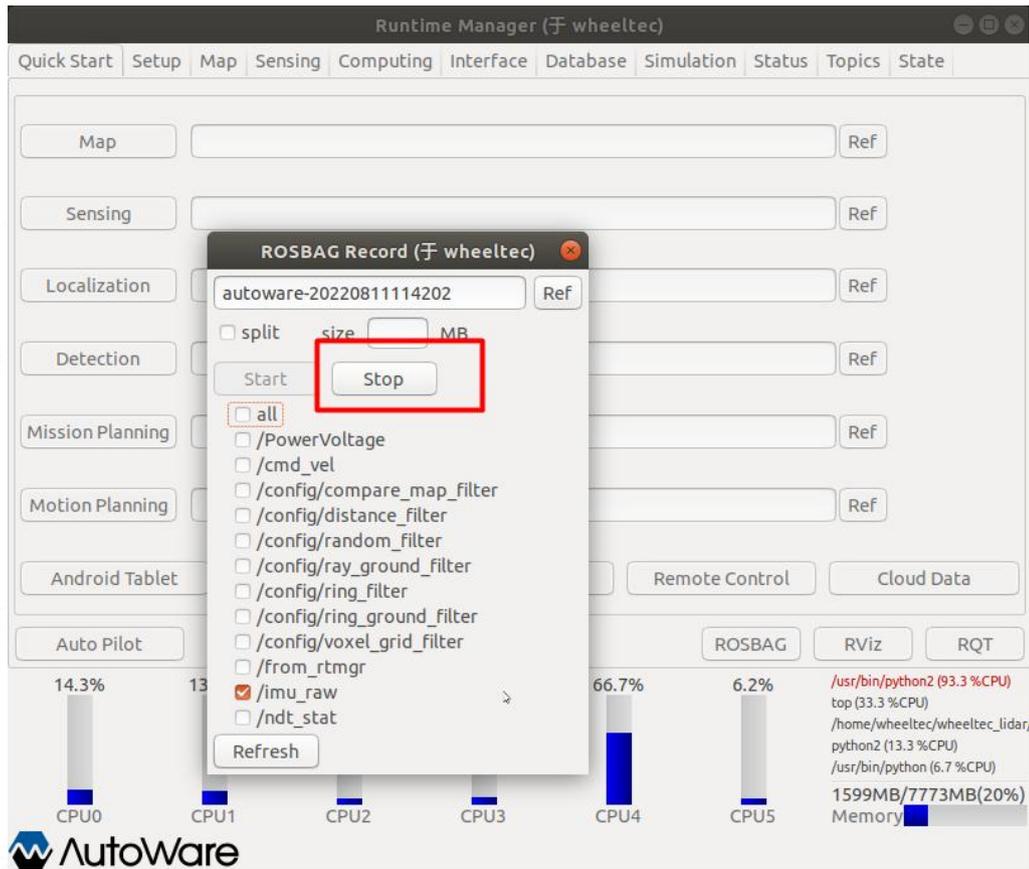


图 2-1-4 Autoware 保存 ROSBAG

## 2.2 使用 ROSBAG 进行建图

Autoware 主要采用了离线的建图方式进行建图，即利用上一节录制好了数据包进行建图。这里，首先介绍一下 Autoware 建图用的坐标系之间的关系，Autoware 用的坐标系有四类：世界坐标系 world，地图坐标系 map，小车坐标系 base\_link，传感器坐标系 velodyne，它们之间的关系如图所示：

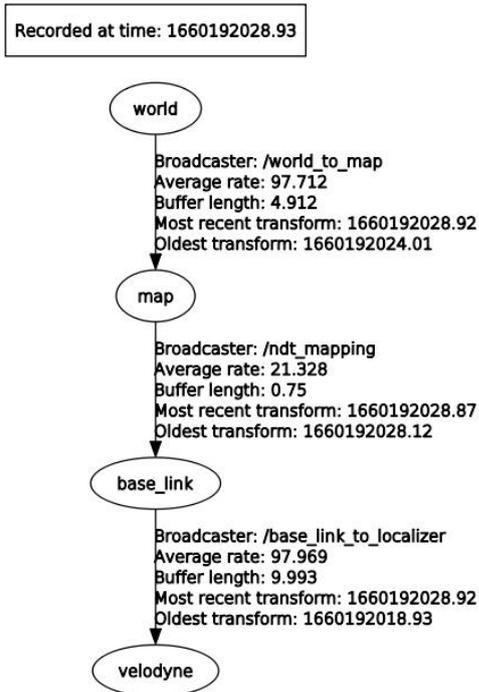


图 2-2-1 Autoware 常见的 TF 关系

从 world 到 map 的坐标系转换与从 base\_link 到 velodyne 的坐标系转换是固定的,用 ROS 的 TF 即可。从 map 到 base\_link 的映射就需要正态分布变换(NDT)算法, autoware 在建图采用的是 ndt\_mapping。

### ①单独打开 Autoware 功能:

建图需要单独打开 autoware, 不能同时运行打开雷达等其他传感器, 雷达等其他传感器信息由 ROSBAG 提供。

- 源码安装方式:

```
roslaunch runtime_manager runtime_manager.launch
```

- Docker 安装方式:

在/home/wheeltec/autoware 目录下运行 sudo ./open.sh 打开 docker 后运行

```
roslaunch runtime_manager runtime_manager.launch
```

运行成功后, 会弹出 runtime\_manager 控制界面:

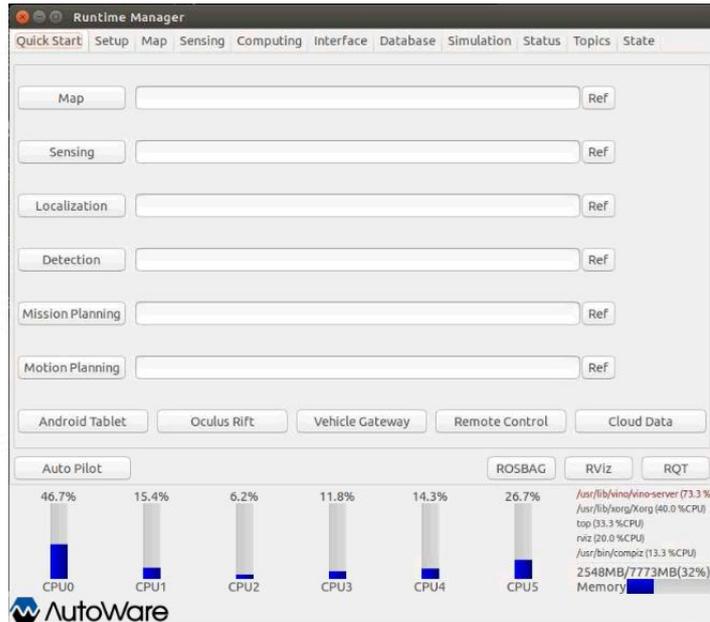


图 2-2-2 Autoware 启动页面

## ②加载点云数据. bag 文件

点击进 [Simulation] 页面, 点击界面右上方 [Ref] 按钮, 加载之前录制的 bag 文件。

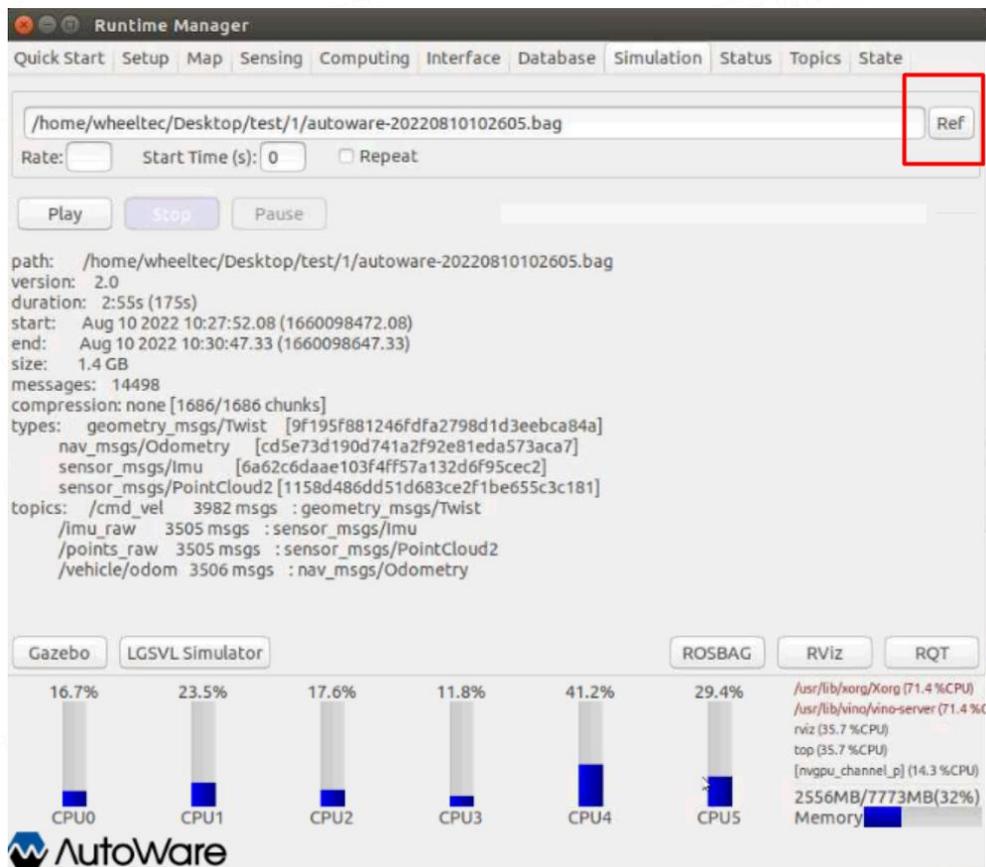


图 2-2-3 Autoware 加载 ROSBAG 包

### ③回放点云数据

点击[Play]按钮开始播放数据，然后点击[Pause]暂停播放。此时你会发现右边进度条有数值出现，如下所示：

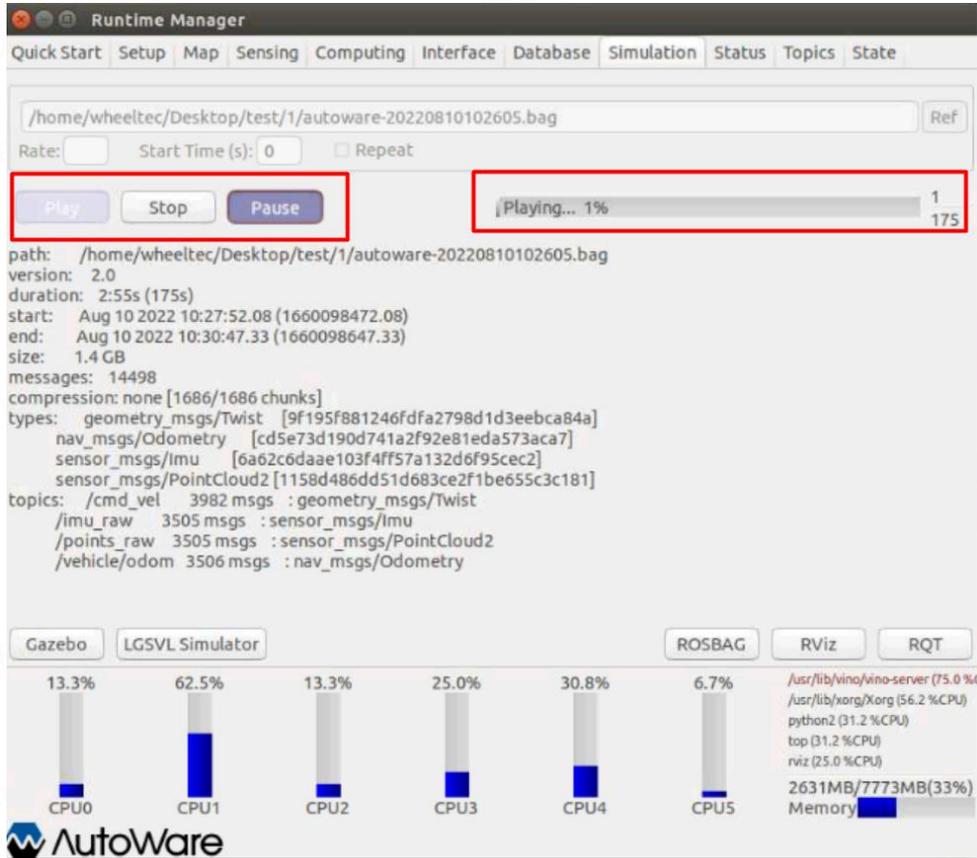


图 2-2-4 回放点云数据

### ④加载从 base\_link 到 velodyne 雷达坐标系的 TF，加载小车模型

点击进入[Setup]页面，确保[Localizer]下选项为[Velodyne]，在[Baselink to Localizer]中设置好各个参数之后点击 TF 按钮，其中 x、y、z、yaw、pitch、roll 表示真车雷达中心点与车身后轴中心点的 TF 位置关系，可根据雷达实际位置进行修改（对于镭神 16 线雷达 yaw 参数可设置成-1.57）。

源码安装方式的可以在 `turn_on_wheeltec_robot` 功能包下选择合适的 urdf 小车模型（docker 版本的小车模型在 `autoware_shared_dir/urdf` 目录下面），然后点击[Vehicle Model]加载模型，如果[Vehicle Model]为空，那么会加载一个默认模型（在 rviz 显示时，如果有激光雷达数据，车辆会显示为黑色）。

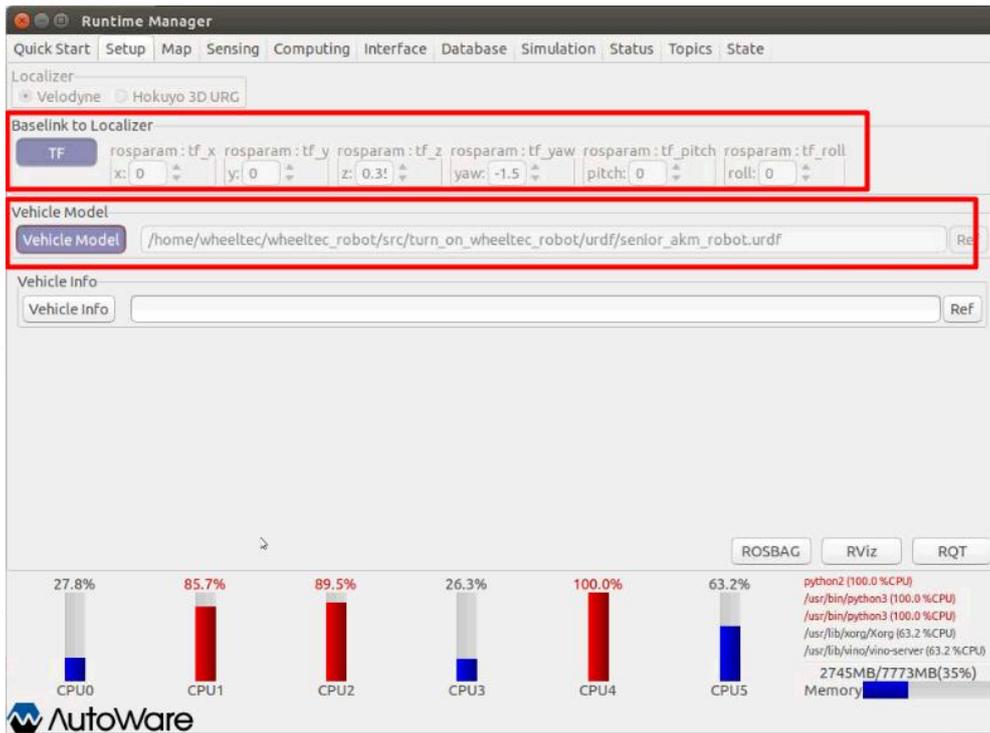


图 2-2-5 加载从 base\_link 到 velodyne 坐标系的 TF，加载小车模型

### ⑤设置从 world 到 map 转换

点击[Map]页面，先点击[TF]的[ref]，选择  
`~/wheeltec_robot/src/turn_on_wheeltec_robot/launch/lgsvl-tf.launch` 文件后（docker  
 安装版本的 lgsvl-tf.launch 文件在~/autoware\_shared\_dir 目录下面），再点击[TF]  
 加载该文件，如图 2-2-6。

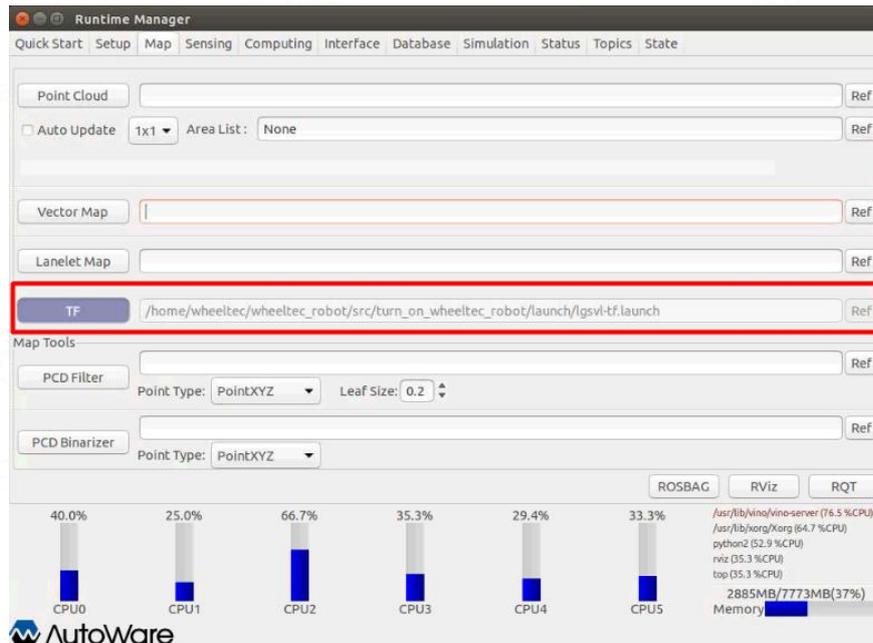


图 2-2-6 设置从 world 到 map 转换

这是加载默认 world 到 map 的坐标转换，打开 lgsvl-tf.launch 文件如下：

```
<launch>
  <node pkg="tf" type="static_transform_publisher" name="world_to_map" args="0 0
0 0 0 /world /map 10" />
</launch>
```

### ⑥设置 map 到 base\_link 转换

在[Computing]菜单栏中找到[lidar\_localizer]下的[Ndt\_Mapping]选项，需要先设置[app]的一些参数后（对 app 内的参数数据修改完成时，需要按回车键，左侧的滑动栏位置发生变化数据才能保存成功），才能勾选[Ndt\_Mapping]。

Ndt\_Mapping 有一些参数设置需要注意一下，如下图：

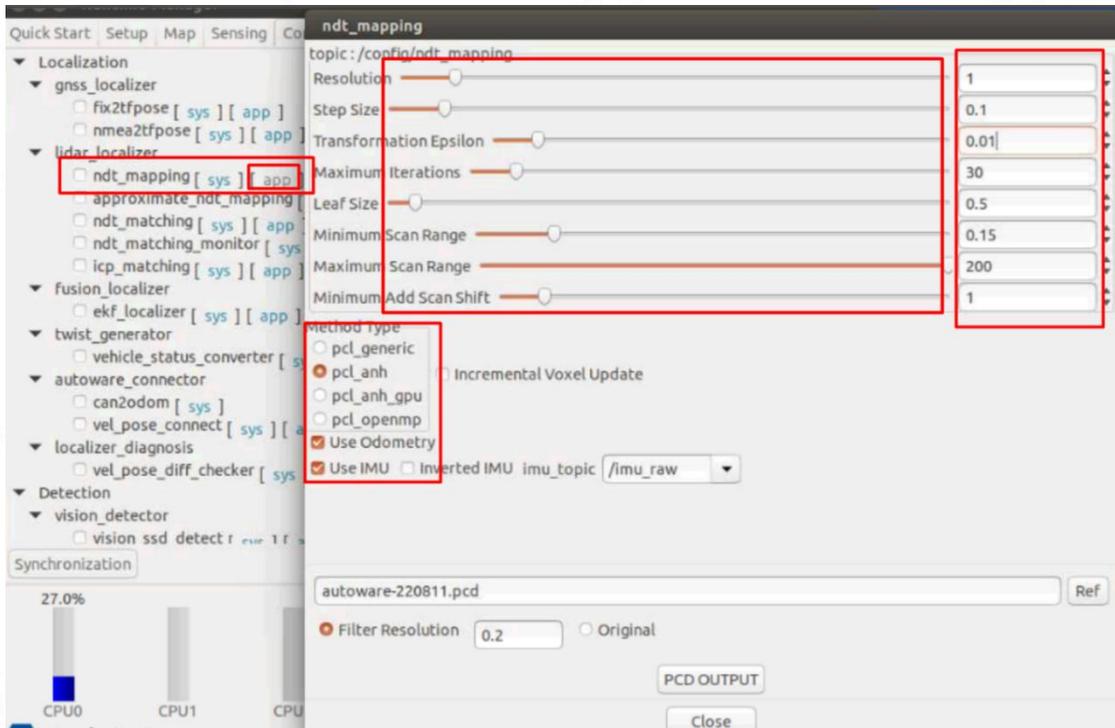


图 2-2-7 设置从 map 到 base\_link 转换

Resolution: ndt 分辨率，最大建图精度越低，默认为 1 即可

Step size: 步长，默认为 0.1

Transformation epsilon: 两次正态变换允许的最大值，默认为 0.01 米

Maximum iterations: 最大迭代次数，默认为 30 次

Leaf size: 点云降采样用的体素叶大小，室内可设置小点，室外可设置大点

**Minimum scan range:** 设置雷达最小范围，过滤较近处雷达点云，推荐设置小点：**0.15 米**

Maximum scan range: 设置雷达最大范围，过滤较远处雷达点云，默认 200 米

在[method type]栏中，勾选[pcl\_anh]选项（选其中一个，最终节点能成功运行即可。运行失败可以尝试勾选其他选项。无 GPU 加速的选[pcl\_generic]，有 GPU 加速的选[pcl\_anh\_gpu]，[Ndt\_Mapping]在本机环境中使用[pcl\_anh\_gpu]会存在一定问题，推荐勾选[pcl\_anh]选项。）。另外，如果要用到里程计或者 imu 辅助定位的，把[use odometry]或者[use imu]选项勾选上。

### ⑦继续播放点云数据包

点击进 [Simulaton]页面，再次点击[Pause]按钮可以继续播放数据，此时命令行打印一些日志信息如下所示：

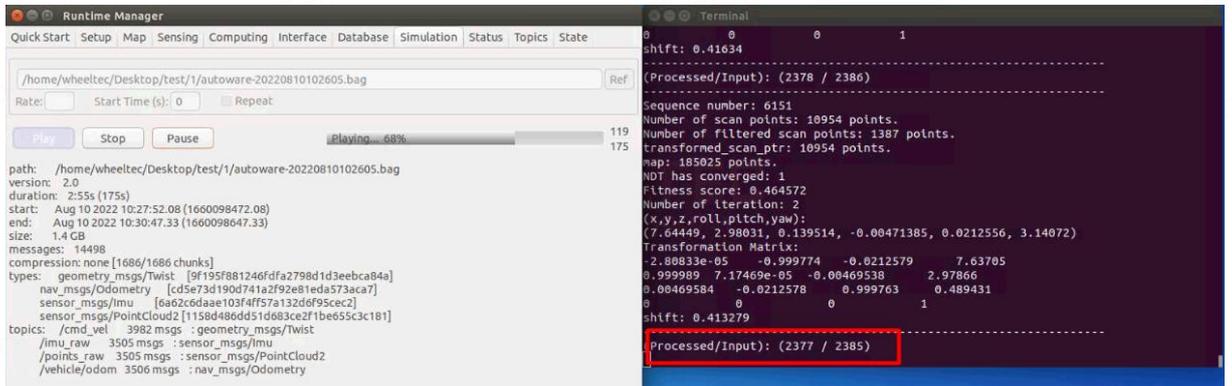


图 2-2-8 继续播放点云数据包

红框位置中，左边数字为已处理的数据，右侧的数字为总的的数据，二者不能相差太大，相差太大程序有可能卡死，推荐相差 200 左右就按下[Pause]按钮暂停播放数据，等待当前数据处理完成就再次点击[Pause]按钮继续播放数据。

### ⑧保存 3D 点云地图

等待点云数据加载完毕，命令行停止打印相关日志时，如图所示，便可保存点云地图。

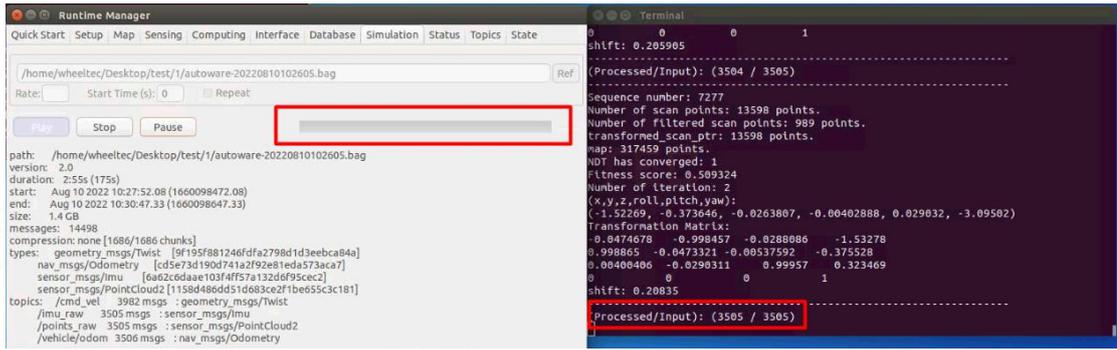


图 2-2-9 点云数据加载完毕

打开[Computing]页面，找到[lidar\_localizer]下的[Ndt\_Mapping]选项，点击[app]，在[ref]中选择点云地图保存的路径后，点击[PCD OUTPUT]即可保存点云地图（注意：docker 安装版本的需要保存在~/autoware\_shared\_dir 目录下面）。

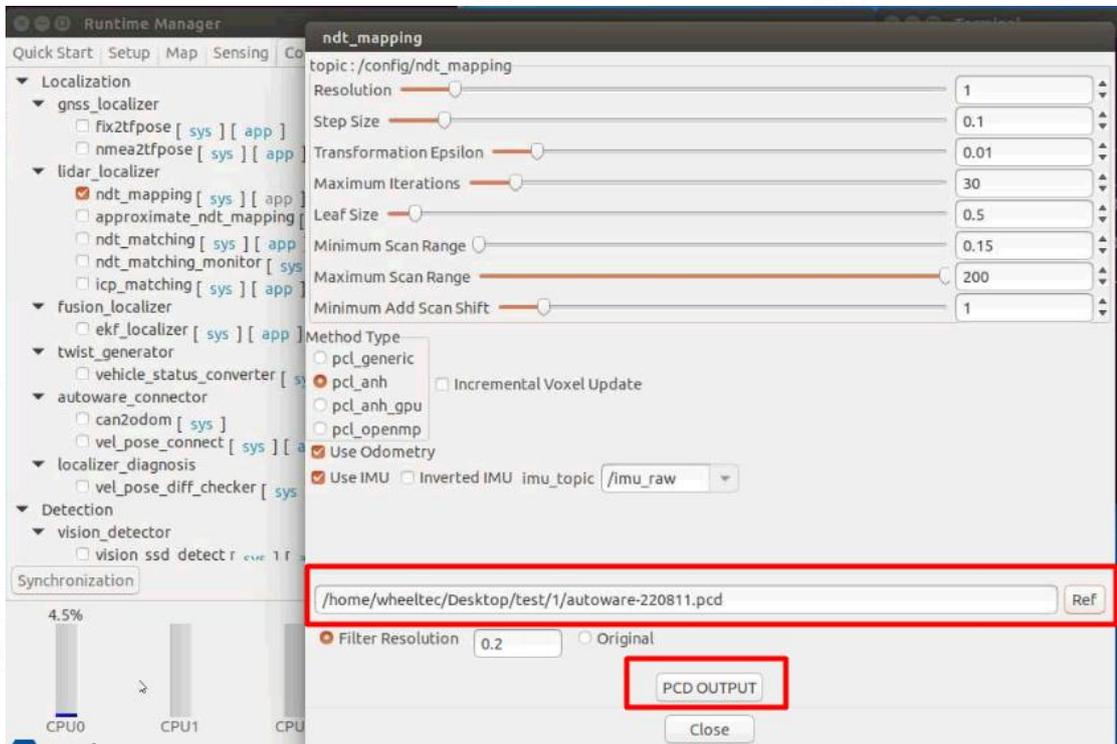


图 2-2-10 保存 3D 点云地图

保存完地图之后，可以通过[pcl\_viewer 点云文件]命令打开点云地图查看当前建图效果。

## 2.3 使用 ROSBAG 生成路径点

路径点文件主要是为了后面 2.6 节和 2.7 节的导航提供导航路径。

①~ ⑤步骤请参考 [2.2 节①~⑤内容](#)

## ⑥ 加载点云地图 PCD 文件

打开[map]页面，点击[point cloud]的[ref]，选择之前建图的 3D 点云地图.pcd 文件，点击[point cloud]按钮，此时下方会出现一个进度条，当进度条显示加载了 100% 并出现[OK]字样时，证明 3D 点云地图已经成功加载完毕，如下图：

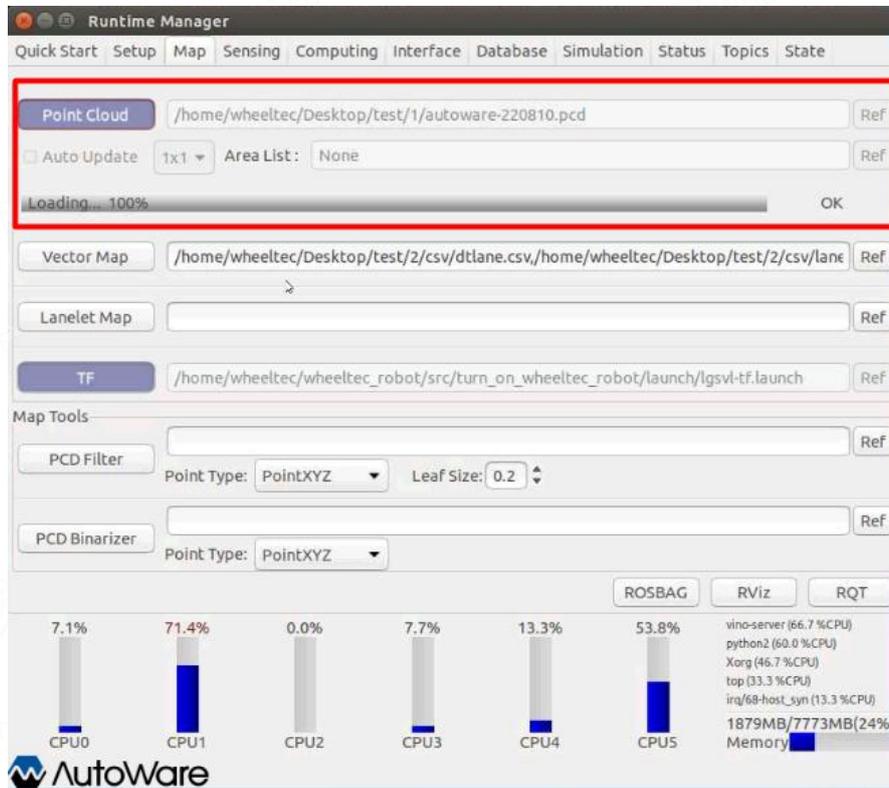


图 2-3-1 加载 3D 点云地图

## ⑦ 点云降采样过滤

打开[sensing]页面，找到[points\_downsampler]下的[voxel\_grid\_filter]选项，设置[app]的一些参数后，勾选[voxel\_grid\_filter]。[voxel\_grid\_filter]有一些参数设置如下图：

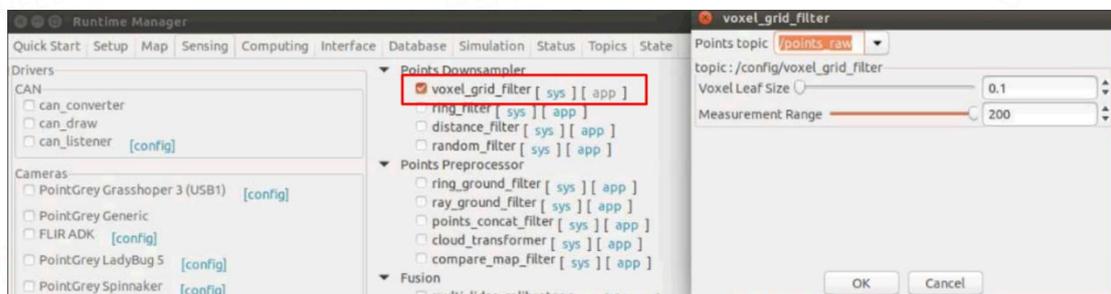


图 2-3-2 点云降采样过滤

Voxel Leaf size: 体素降采样算法中的体素叶大小，室内可设置小点，室外

可设置大点

### ⑧ 设置从 map 到 base\_link 的转换 (NDT\_MATCHING)

找到[Computing]左菜单栏下的[ndt\_matching]选项, 设置[app]里面的参数后, 勾选[ndt\_matching], 一些参数设置如下所示:

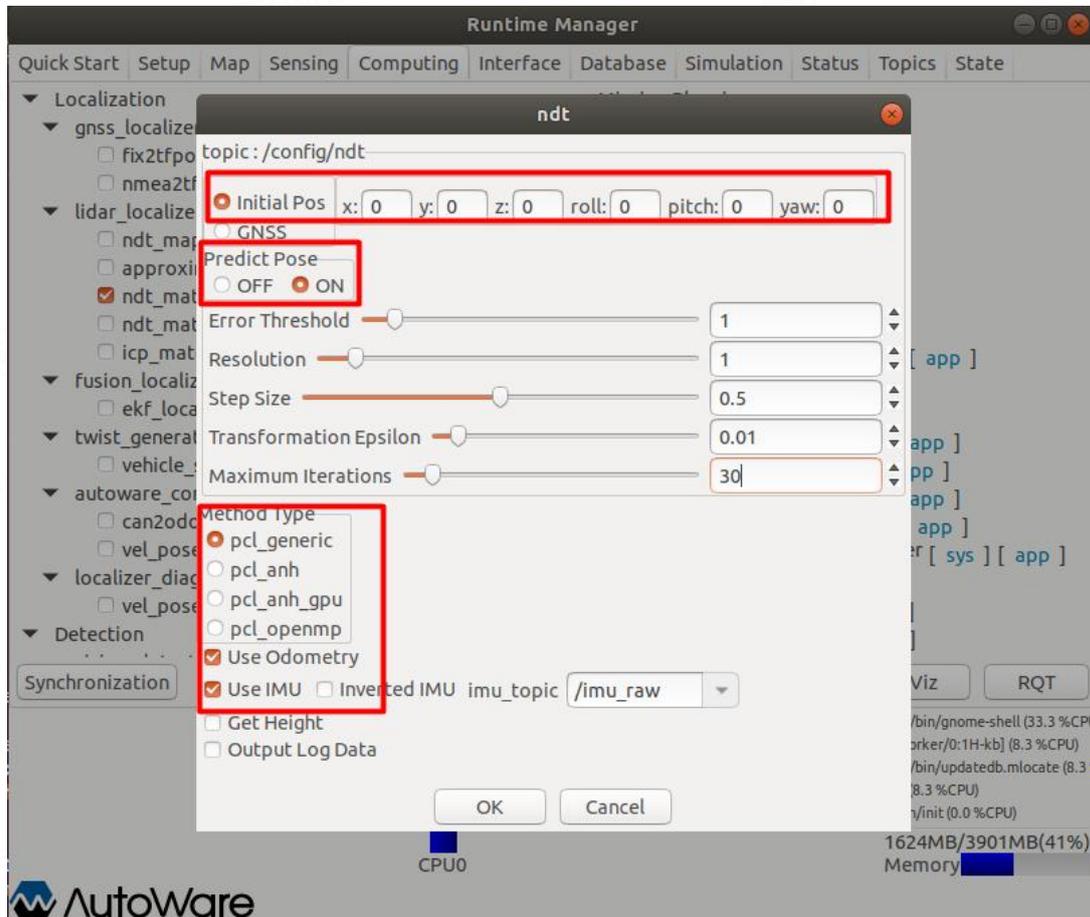


图 2-3-3 设置从 map 到 base\_link 的转换

这里没有使用 GNSS, 所以[topic:/config/ndt] 选项处于 [Initial\_Pose] 处, 勾选 [Initial Pos], x, y, z, roll, pitch, yaw 的值表示小车的初始位置 (默认建图起点位置)。

在[method type]栏中, 无 GPU 加速的选[pcl\_generic], 有 GPU 加速的选 [pcl\_anh\_gpu]。另外, 本文要用到里程计和 imu 辅助定位, 把[use odometry]和[use imu]选项勾选上, 其余参数参考默认值就可以了。

### ⑨ 启动[vel\_pose\_connect]

找到 [Computing] 左菜单栏下的 [vel\_pose\_connect], 打开 [app] 并确保选

项 [Simulation\_Mode] 没有被勾选，退出并勾选 [vel\_pose\_connect]。

### ⑩ 设置路径点文件保存路径

找到 [Computing] 右菜单栏下的 [waypoint\_saver]，打开 [app]，点击[ref]选择保存的路径(注意: docker 版本的需要保存在~/autoware\_shared\_dir 目录下面)，勾选上[save /current\_velocity]，退出并勾选[waypoint\_saver]。

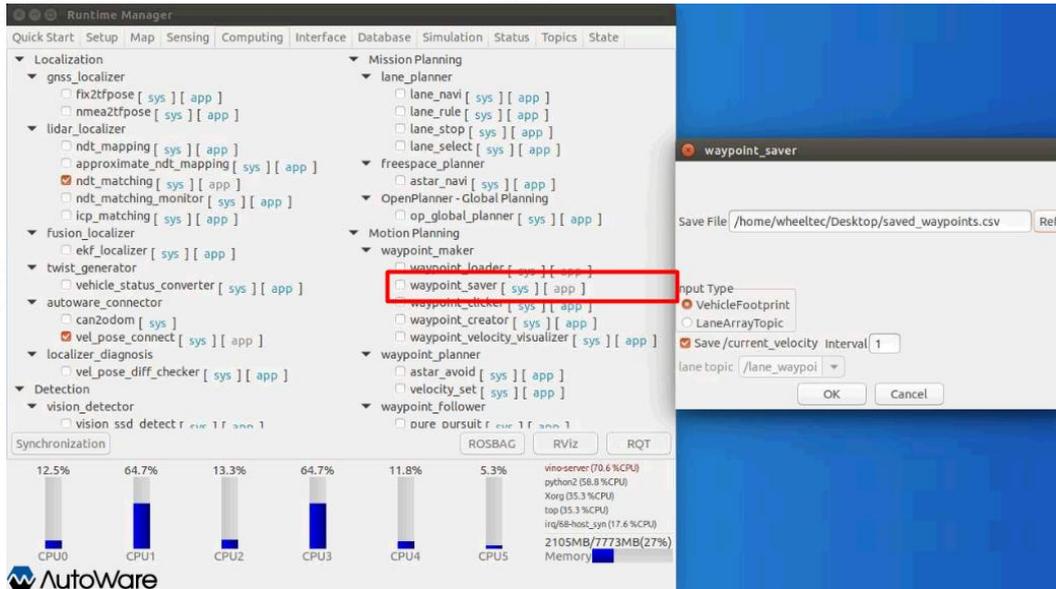


图 2-3-4 设置路径点文件保存路径

### ⑪ 继续播放点云数据包

点击进入 [Simulaton] 页面，再次点击[Pause]按钮可以继续播放数据，等待数据包回放结束便可，路径点文件会自动保存在设置好的路径下。

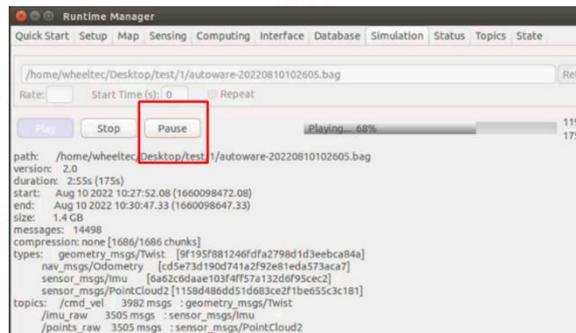


图 2-3-5 继续播放点云数据包

### ⑫ rviz 查看路径点

在路径点生成过程中，源码安装方式的可以点击界面右下角的[rviz] 打开 rviz，相关配置在

~/autoware.ai/src/autoware/documentation/autoware\_quickstart\_examples/launch/rosbag\_demo/default.rviz 中（docker 版本的需要在小车上运行 rviz 单独打开，相关配置在~/autoware\_shared\_dir/default.rviz 中）。此时我们可以看到 rviz 中逐渐出现一条路径：

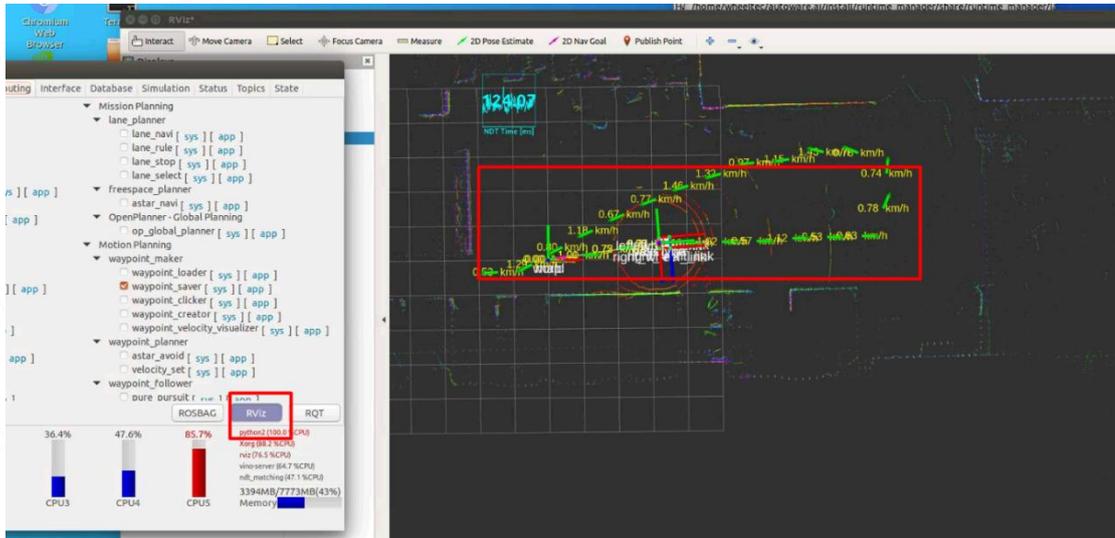


图 2-3-6 rviz 查看路径点

## 2.4 地图定位

①运行以下命令启动雷达、STM32、Autoware:

- 源码安装方式:

启动雷达、STM32、Autoware:

```
roslaunch turn_on_wheeltec_robot open_autoware.launch
```

- Docker 安装方式:

启动雷达、STM32:

```
roslaunch turn_on_wheeltec_robot open_autoware.launch
```

启动 autoware:

在/home/wheeltec/autoware 目录下运行 sudo ./open.sh 打开 docker 后运行

```
roslaunch runtime_manager runtime_manager.launch
```

②加载从 base\_link 到 velodyne 坐标系的 TF，加载小车模型

点击进入[Setup]页面，确保[Localizer]下选项为[Velodyne]，在[Baselink to Localizer]中设置好各个参数之后点击 TF 按钮，其中 x、y、z、yaw、pitch、roll 表示真车雷达中心点与车身后轴中心点的 TF 位置关系，可根据雷达实际位置进

行修改（对于镭神 16 线雷达 yaw 参数可设置成-1.57）。

此时可以在 turn\_on\_wheeltec\_robot 功能包下选择合适的 urdf 小车模型（docker 安装方式的在~/autoware\_shared\_dir/urdf中），点击[Vehicle Model]加载模型，如果[Vehicle Model]为空，那么会加载一个默认模型（在 rviz 显示时，如果有激光雷达数据，车辆会显示为黑色）。

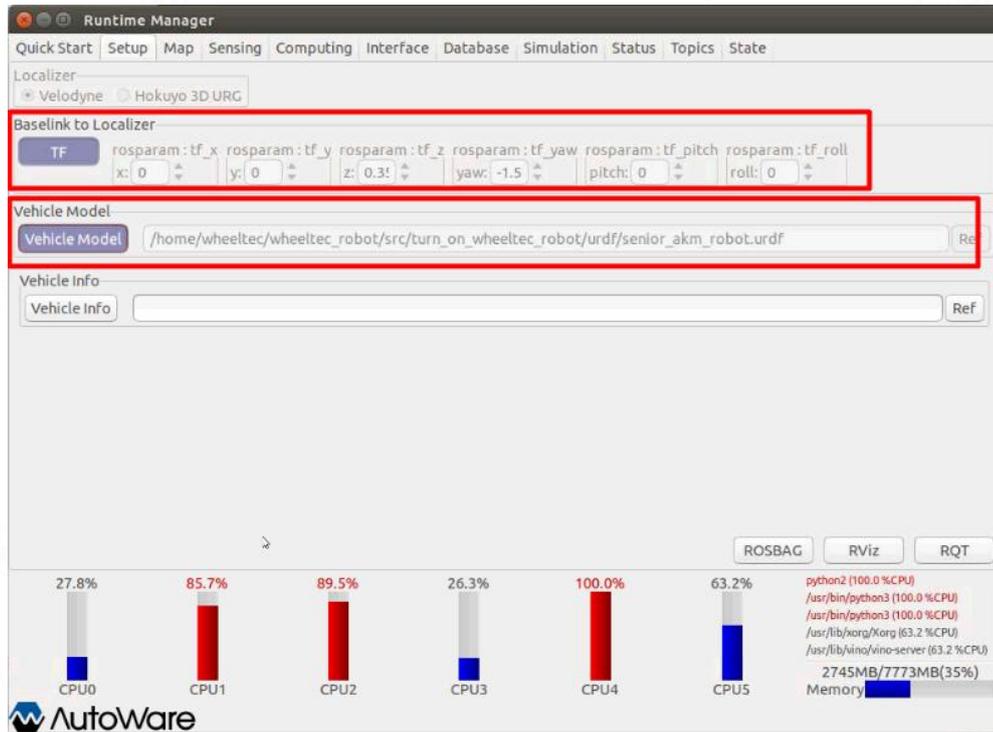


图 2-4-1 加载从 base\_link 到 velodyne 坐标系的 TF，加载小车模型

### ③设置从 world 到 map 转换

点击[Map]页面，点击[TF]的[ref]选择，源码安装方式的选择~/wheeltec\_robot/src/turn\_on\_wheeltec\_robot/launch/lgsvl-tf.launch 文件（docker 安装方式的选择~/autoware\_shared\_dir/lgsvl-tf.launch 文件），这是加载默认 world 到 map 的坐标转换。

### ④加载点云地图 PCD 文件

打开[map]页面，点击[point cloud]的[ref]，选择之前建图的 3D 点云地图.pcd 文件，点击[point cloud]按钮，此时下方会出现一个进度条，当进度条显示加载了 100%并出现[OK]字样时，证明 3D 点云地图已经成功加载完毕，如下图：

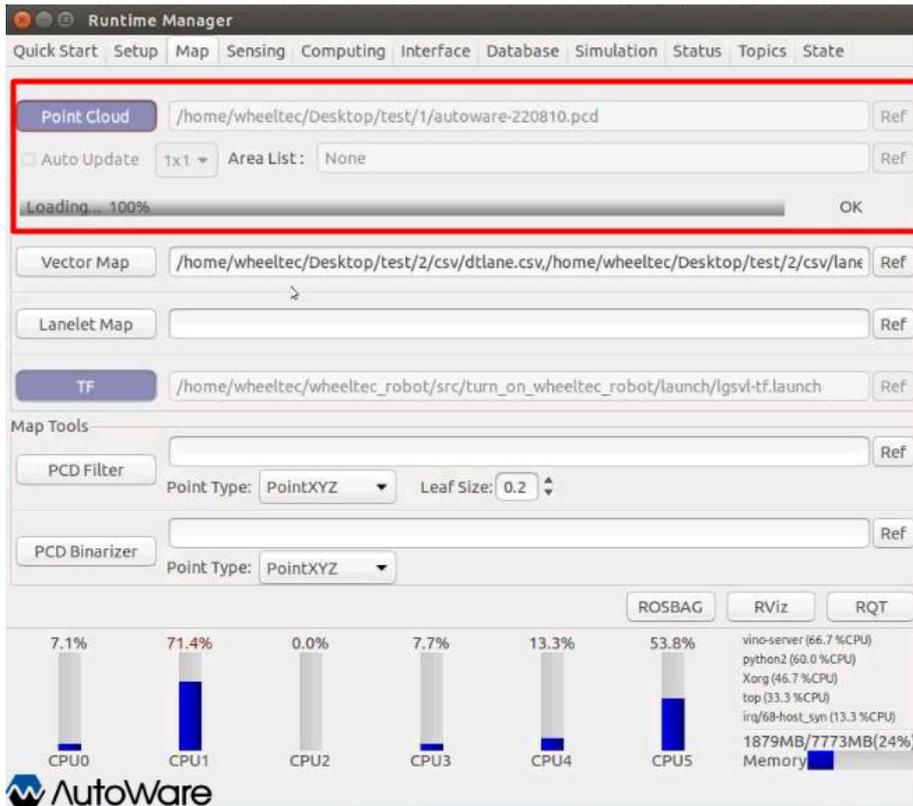


图 2-4-2 加载点云地图 PCD 文件

### ⑤点云降采样过滤

打开[sensing]页面，找到[points\_downsampler]下的[voxel\_grid\_filter]选项，设置[app]的一些参数后，并勾选。[voxel\_grid\_filter]有一些参数设置如下图：

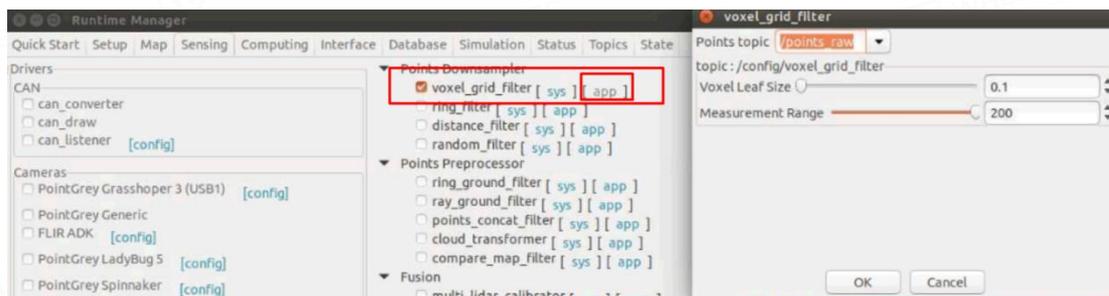


图 2-4-3 点云降采样过滤

Voxel Leaf size: 体素降采样算法中的体素叶大小，室内可设置小点，室外可设置大点

### ⑥地面过滤

打开[sensing]页面，找到[points\_preprocessor]下的[ring\_ground\_filter]选项，设置[app]的一些参数后，并勾选。[ring\_ground\_filter]有一些参数设置如下图：

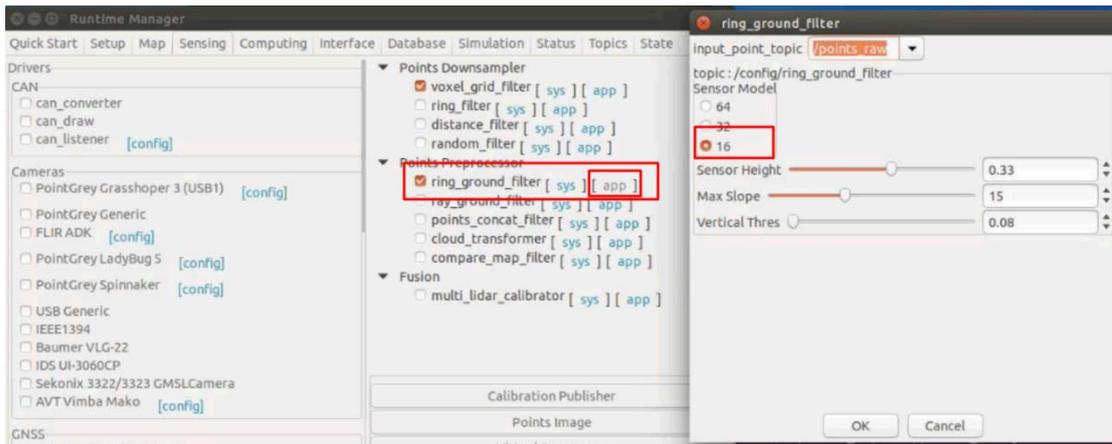


图 2-4-4 地面过滤

**Sensor model:** 16、32、64 线雷达，这里如果使用镭神 16 线雷达勾选[16]

**Sensor height:** 雷达距离地面高度，根据实际情况来填写即可

**Max slope:** 地面最大坡度，单位： $^{\circ}$ ，如果不给出坡度，容易将坡识别为障碍物

**Vertical Thres:** 表示障碍物和地面的差异度，如果大于这个值则被认为是障碍物

#### ⑦设置从 map 到 base\_link 的转换 (NDT\_MATCHING)

找到[Computing]左菜单栏下的[ndt\_matching]选项，设置[app]后，勾选上[ndt\_matching]，其中一些参数设置如下所示：

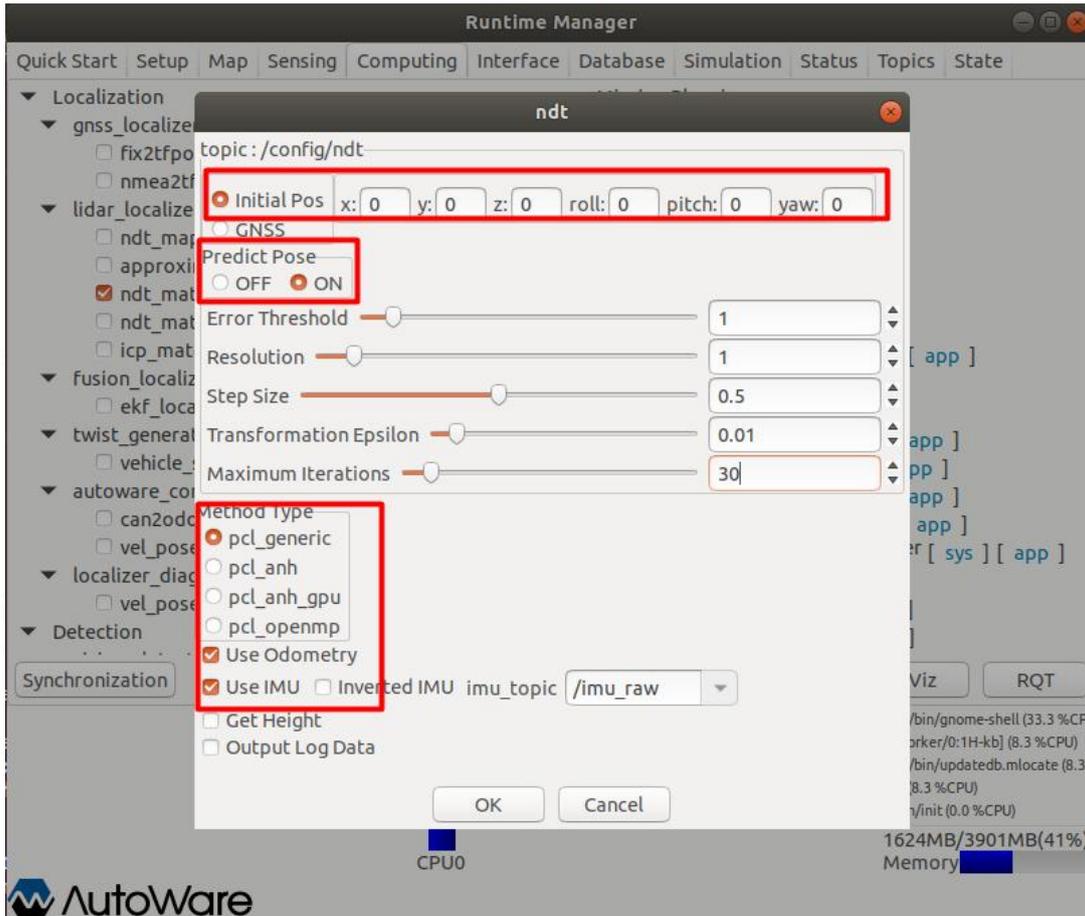


图 2-4-5 设置从 map 到 base\_link 的转换

这里没有使用 GNSS，所以[topic:/config/ndt]选项处于 [Initial\_Pose] 处，勾选 [Initial Pos]，x, y, z, roll, pitch, yaw 的值表示激光的初始位置（即建图起点位置）。

在[method type]栏中，选其中一个，最终节点能成功运行即可。运行失败可以尝试勾选其他选项。无 GPU 加速的选[pcl\_generic]，有 GPU 加速的选 [pcl\_anh\_gpu]。另外，本文要用到里程计和 imu 辅助定位，把[use odometry]和[use imu]选项勾选上，其余参数参考默认值就可以了。

### ⑧启动[vel\_pose\_connect]

找到 [Computing] 左菜单栏下的 [vel\_pose\_connect]，打开 [app] 并确保选项 [Simulation\_Mode] 没有被勾选，退出并勾选 [vel\_pose\_connect]。

### ⑨打开 rviz:

源码安装方式的可以点击界面右下角的[rviz] 打开 rviz，相关配置在 [~/autoware.ai/src/autoware/documentation/autoware\\_quickstart\\_examples/launch/ros](~/autoware.ai/src/autoware/documentation/autoware_quickstart_examples/launch/ros)

bag\_demo/default.rviz 中（docker 版本的需要在小车上运行 rviz 单独打开，相关配置在~/autoware\_shared\_dir/default.rviz 中）

若是小车不在建图起点位置，或者是定位不准确，可以点击 rviz 菜单栏中的 [2D pose estimate]，然后在地图上鼠标左击，拖动，重新设置小车的位置。

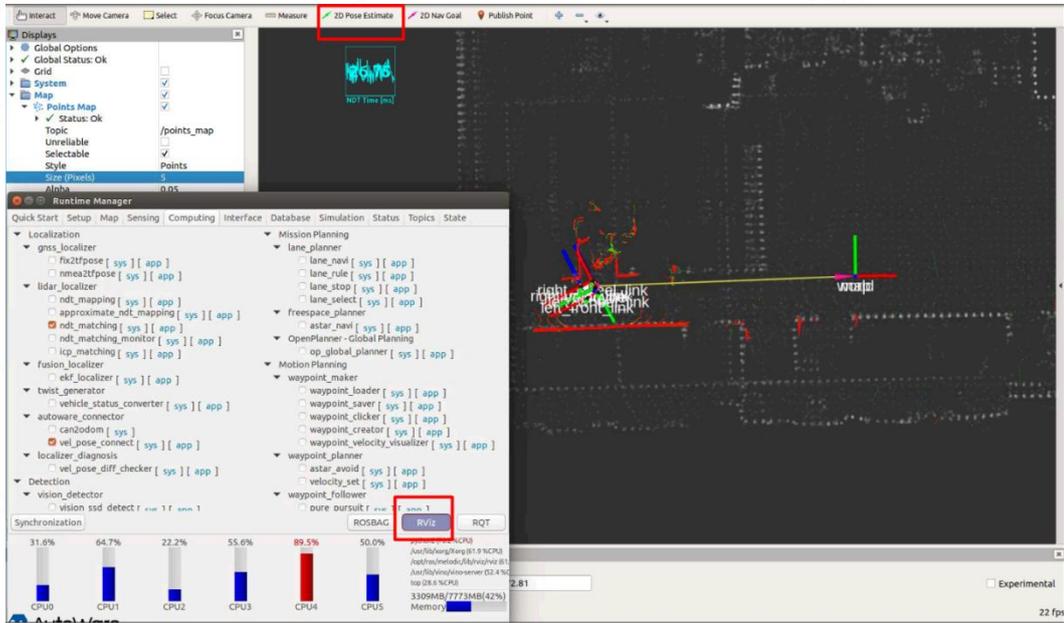


图 2-4-6 打开 rviz

## 2.5 雷达目标识别

①~⑨步骤请参考 [2.4 节①~⑨内容](#)

### ⑩ 欧几里得聚类

打开[Computing]页面，找到[lidar\_detector]下的[lidar\_euclidean\_cluster\_detect]选项，设置[app]的一些参数后，并勾选。[lidar\_euclidean\_cluster\_detect]有一些参数设置如下图：

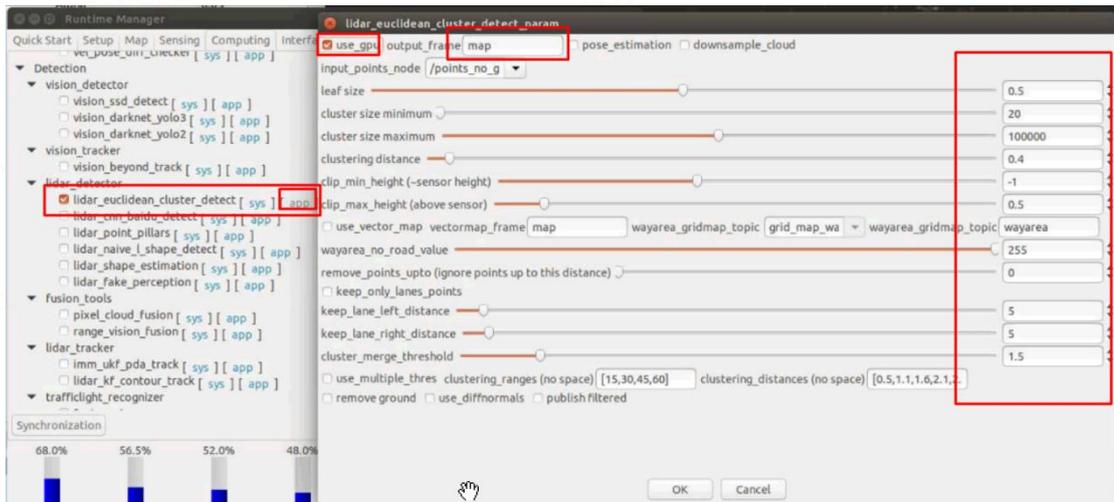


图 2-6-1 欧几里得聚类

**Use\_gpu:** 是否使用 GPU 加速，有 GPU 加速的勾选上即可

**Output\_frame:** 需要修改成 map 才能实现避障效果

**Input\_point\_node:** 雷达话题，修改选中过滤地面后的雷达话题  
/points\_no\_ground

**Leaf size:** 点云降采样用的体素叶大小，室内可设置小点，室外可设置大点

**Cluster size minimum:** 聚类所需要的最小点数要求，太小容易识别噪声点为一个聚类目标结果，太大容易忽略某些小物体

**Clustering distance:** 不同聚类结果的最小间隔

**Clip\_min\_height/Clip\_max\_height:** 只保留

**Clip\_min\_height~Clip\_max\_height (m)** 高度之间的点云，其他点云都会被过滤掉

**Remove\_points\_upto (m):** 小于此范围内的点云也会过滤掉，推荐设置为  
0

其他参数默认即可。

### ⑪ 跟踪障碍物

打开[Computing]页面，找到[lidar\_tracker]下的[lidar\_kf\_contour\_track]选项，设置[app]的一些参数后，勾选[lidar\_kf\_contour\_track]。[lidar\_kf\_contour\_track]有一些参数设置如下图：

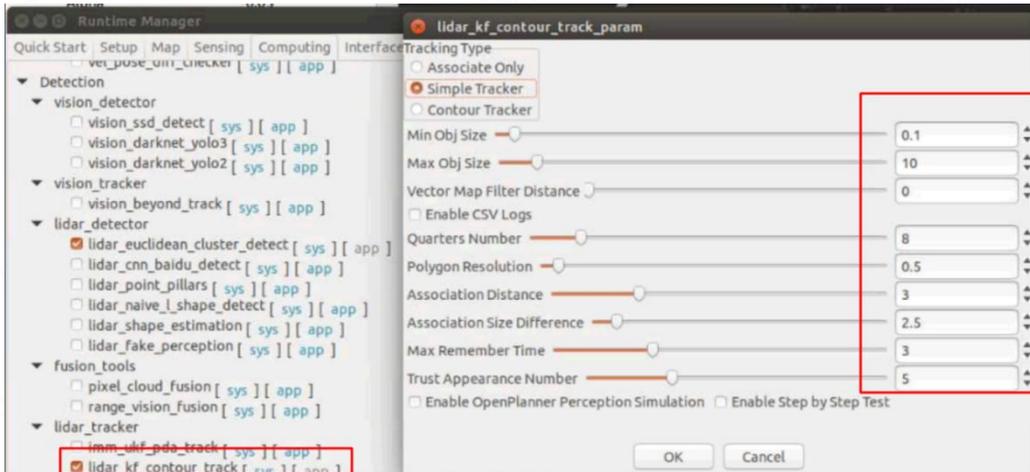


图 2-6-2 跟踪障碍物

Vector map distance: 需要修改为 0

其他参数默认即可。

我们可以通过 rviz 订阅 /detection/lidar\_detector/objects\_markers 话题可以看到聚类及跟踪障碍物得到的结果，也可以通过图片中的步骤订阅 /bounding\_boxes 话题使结果以方框的形式显示出来。

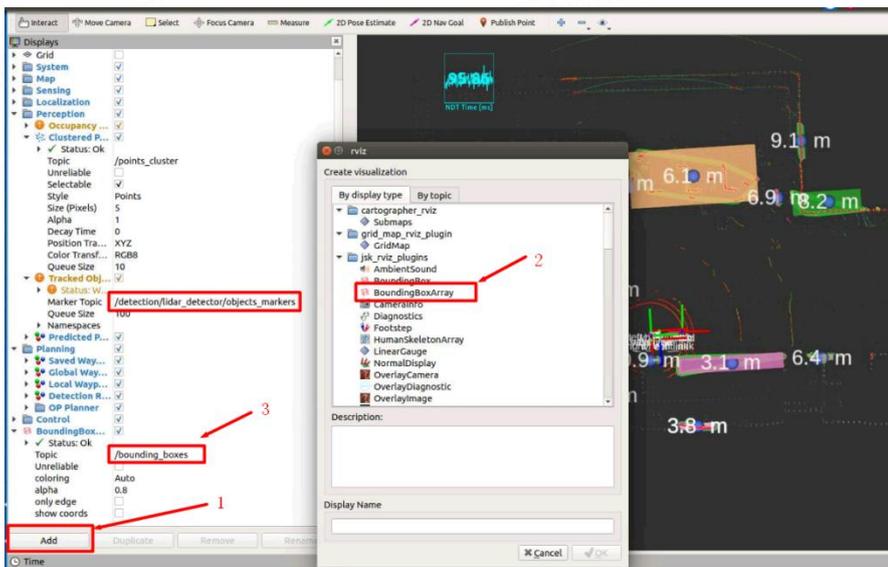


图 2-6-3 欧几里得聚类相关话题订阅

## 2.6 open\_planner 导航

open\_planner 导航主要是利用高精地图进行导航。

### ①完成 2.5 节的雷达目标识别功能

### ②加载高精地图

高精地图地图具体制作方法参考第 3 章内容。

打开[map]页面，点击[vector map]的[ref]，选中全部的高精地图文件，点击[vector map]加载高精地图，此时可以在 rviz 中看到我们画的高精地图。

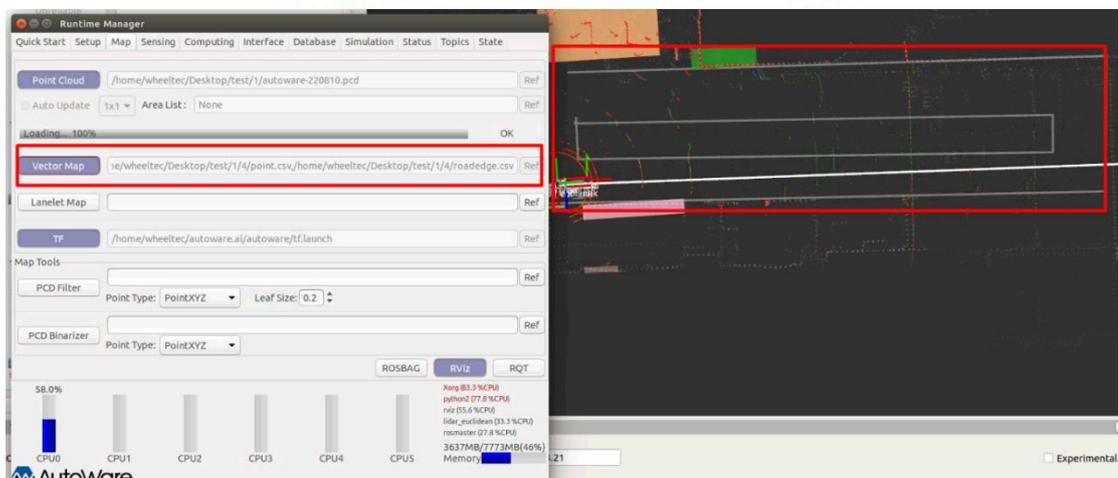


图 2-6-1 加载高精地图

### ③全局路径规划

打开[Computing]页面，找到[openplanner\_global planning]下的[op\_global\_planner]选项，设置[app]的一些参数后，并勾选。然后在 rviz 中点击[2D nav goal]，选择你所要导航的目标点位置（小车起点位置和设定的目标点位置需要放在行驶线 lane 附近）rviz 中会出现一条蓝色路径即为全局路线。

[op\_global\_planner]有一些参数设置如下图：

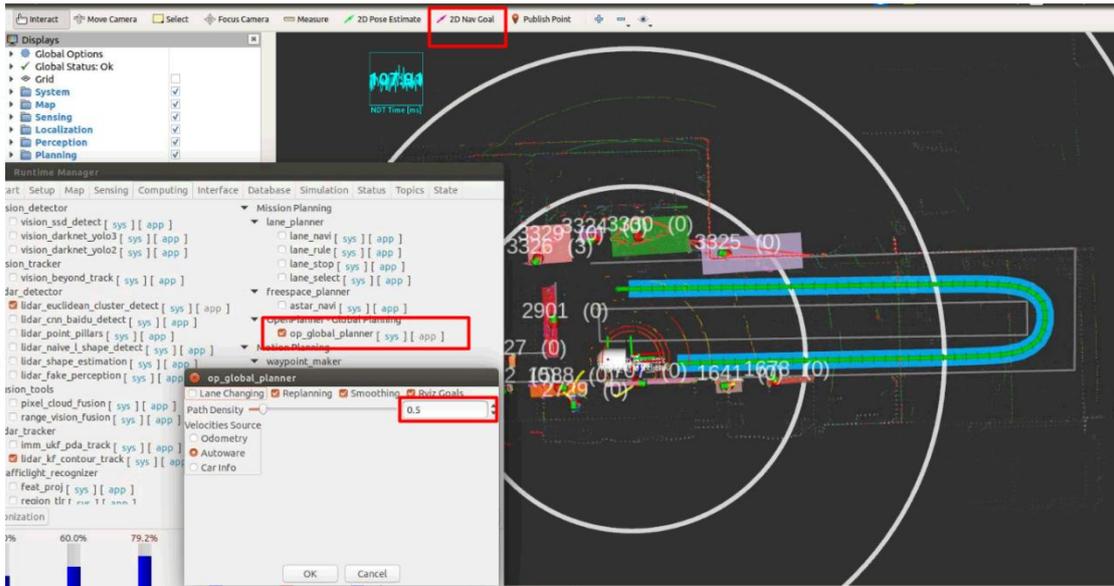


图 2-6-2 全局路径规划

Path density: 由每一个小点组合成全局路径, 此参数为每一个点的间隔距离。

#### ④局部路径规划

打开[Computing]页面, 找到[openplanner\_local planning]栏目, 设置[app]的一些参数后, 需要依次勾选[openplanner\_local planning]栏目下的全部节点。可以在rviz 中点击[add], 添加[local trajectories\_eval\_rviz]话题, rviz 中会出现几条路径即为可行驶的几条局部路线, 红色为不可行驶的路线, 粉红色为当前选择行驶的局部路线, 黄色为可行驶的路线。相关一些参数设置如下图:

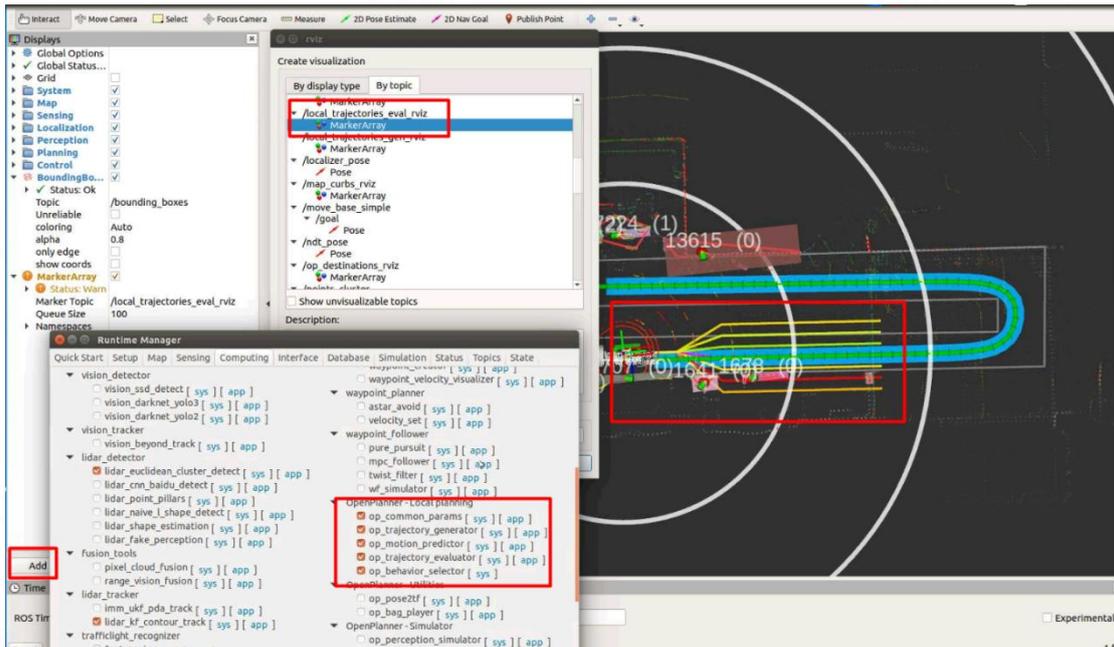


图 2-6-3 局部路径规划勾选节点

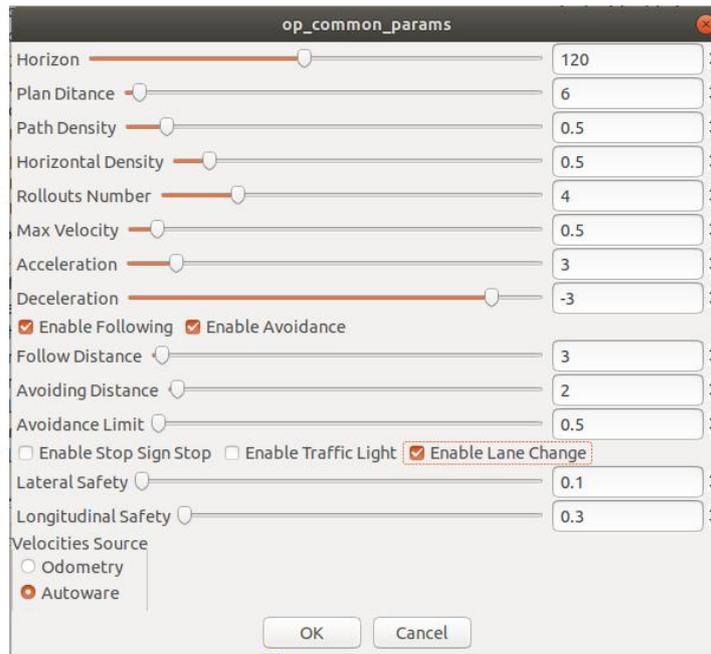


图 2-6-4 op\_common\_params 节点相关参数

**Path distance:** 用于设置主路径两侧的衍生出的局部轨迹的长度

**Path Density** 用于设置一条局部轨迹上两个轨迹点的距离

**Horizontal Density:** 用于设置两个局部轨迹的间距

**Rollouts Number:** 用于设置局部轨迹的数量

**Max velocity:** 小车的行驶速度，不在中心轨迹上的小车行驶速度减半，单位 m/s

**Accelration/deceleration:** 加速度/减加速度

**Avoiding distance:** 距离障碍物多远才开始避障

**Avoidance Limit:** 停车时，障碍物距离多远可以开车

**Lateral Safety** 与 **Longitudinal Safty:** 设置安全框尺寸，分别代表车辆安全框宽与长，障碍物在小车安全框内时，小车停止运动

**Enable following** 和 **enable avoidance:** 需要勾选上，打开跟随和避障功能



图 2-6-5 op\_trajectory\_generator 相关参数

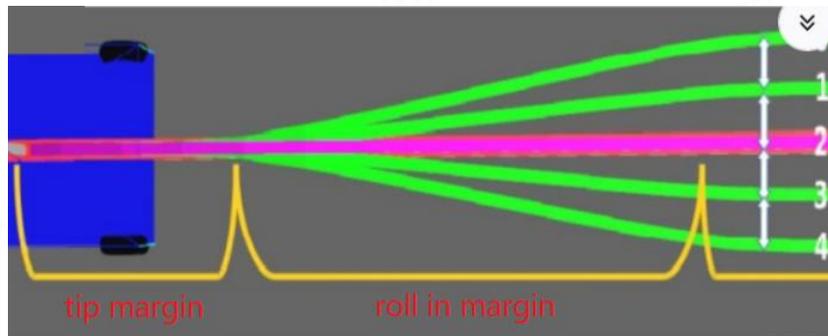


图 2-6-5 Tip Margin 与 Roll In Margin 代表意义

**Tip Margin:** 用于设置车身到路径分叉点的距离

**Roll In Margin:** 用于设置局部轨迹弯折的距离

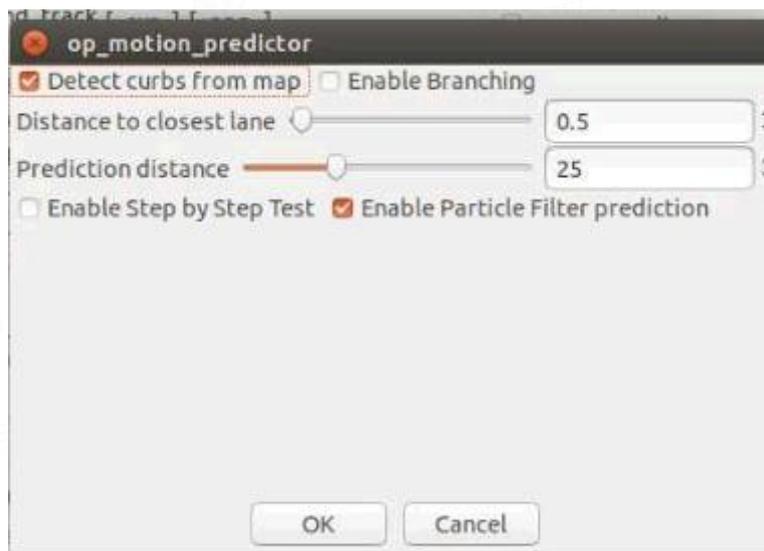


图 2-6-6 op\_motion\_predictor 参数

Detect curbs from map: 用于将路沿判定为障碍物



图 2-6-7 op\_trajectory\_evaluator 相关参数

Enable prediction: 打开预测功能

### ⑤速度输出，小车开始导航

打开[Computing]页面，找到 [waypoint\_follower] 下 [pure\_pursuit] 与 [twist\_filter]，设置[app]的一些参数后，勾选 [pure\_pursuit] 与 [twist\_filter] 。相关一些参数设置如下图：

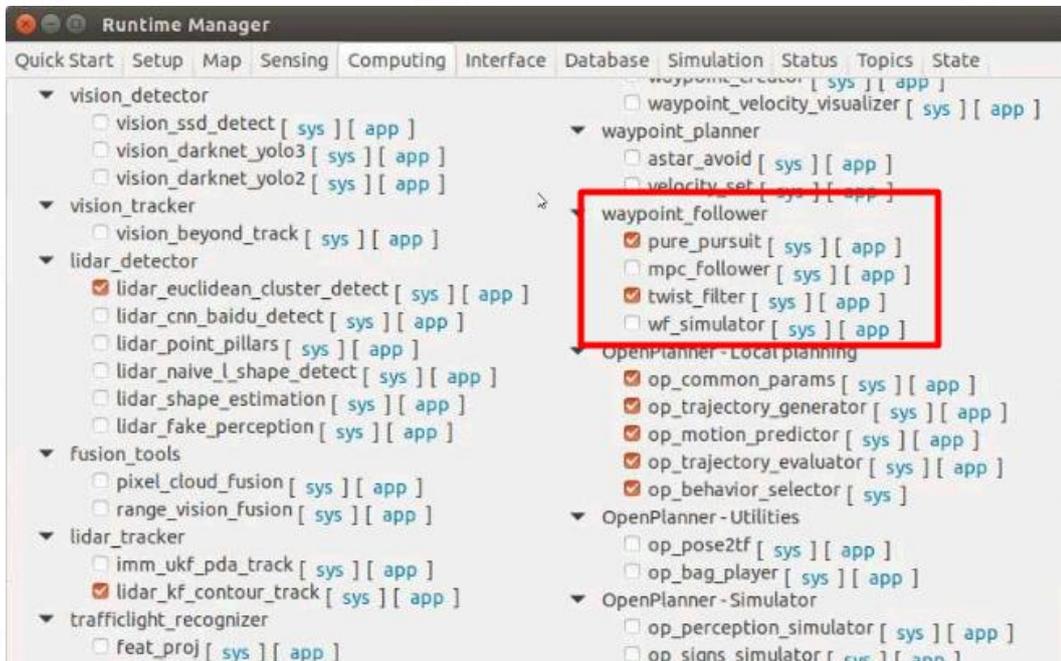


图 2-6-8 速度输出勾选节点

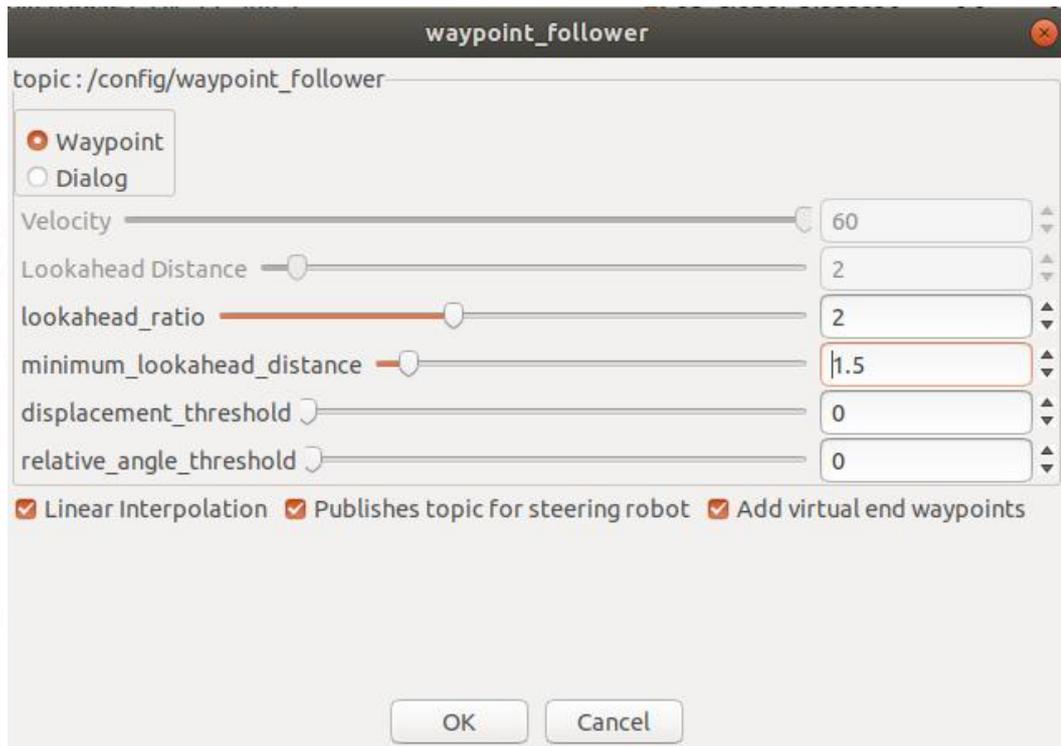


图 2-6-9 pure\_pursuit 相关参数

勾选 waypoint 选项，使用路径点的速度进行输出

Lookahead\_ratio: 前视距离系数

Minimum\_lookahead\_distance: 最小前视距离限制，这里设置为小车的最小转弯半径

Linear interpolation: 是否线性插值

小车的预瞄点距离小车的距离是小车的前视距离，预瞄点是小车在导航中一直在“追”的点，当前行驶速度乘以系数 Lookahead\_ratio 即为小车的前视距离。

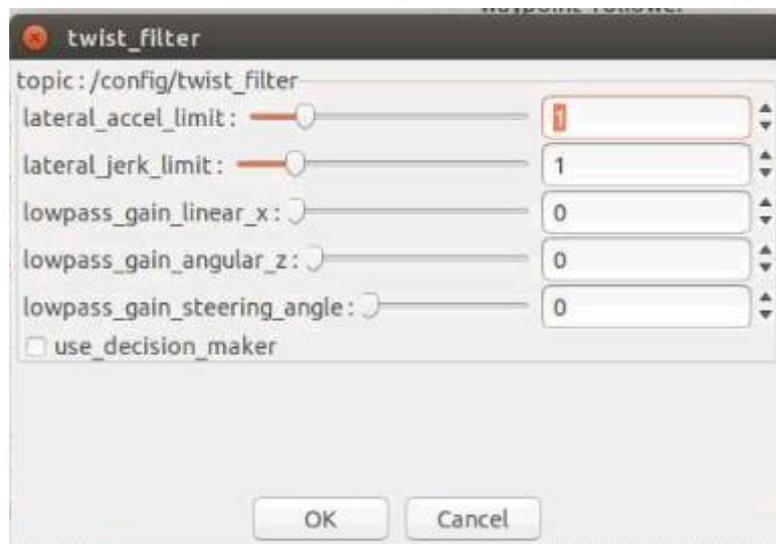


图 2-6-9 twist\_filter 相关参数

Lateral\_accel\_limit 和 Lateral\_jerk\_limit: 横向加减速限制, 默认即可

\*注: 若是导航中小车定位容易乱, 请运行 `rostopic hz /points_raw` 观察小车雷达频率多少, 推荐雷达频率为 20hz 左右, 太小如 10hz 小车运动时容易定位偏移。

## 2.7 路径点导航

此导航功能与 2.6 功能相似, 其中唯一不同的地方是, 第③步骤中全局路径选择在第 2.3 节中生成的路径点文件。

①~②步骤参考[第 2.6 节①~②内容](#)

③ 加载路径点文件

在菜单栏 [Computing] 右侧的 [waypoint\_maker] 下的 [waypoint\_loader] 加载 waypoints 文件, 选择 [Ref] 加载之前保存下来的路点文件, 并点击 [Close] 关闭页面。勾选 [waypoint\_loader], 如下图所示:

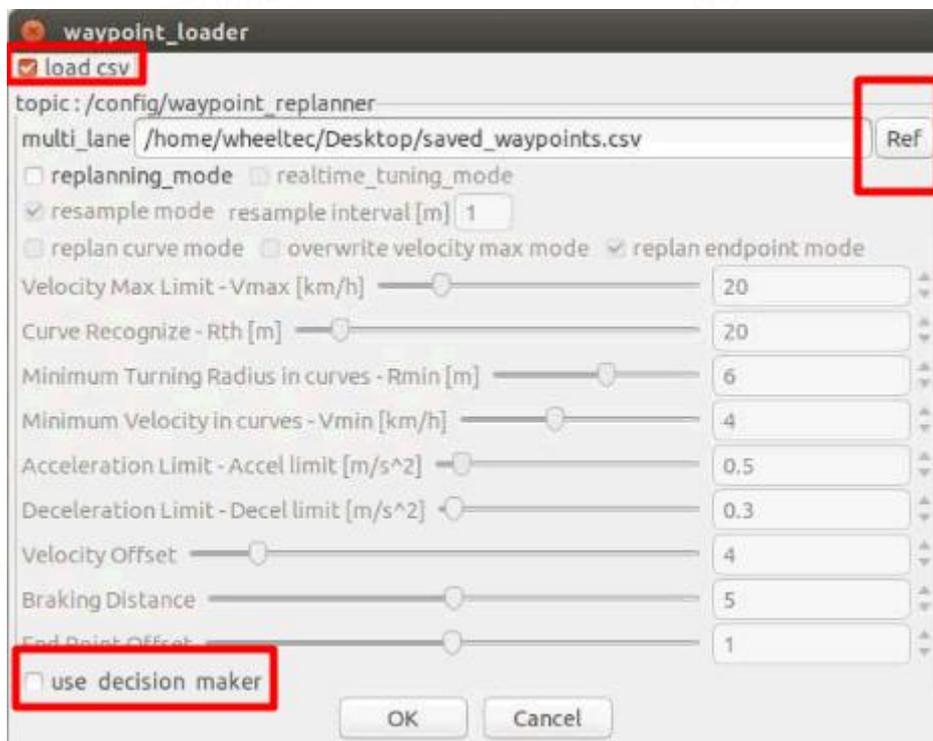


图 2-7-1 加载路径点文件

use\_decision\_maker: 取消勾选

④ ~⑤步骤参考[第 2.6 节④~⑤内容](#)

## 2.8 寻迹

①完成 [2.4 节的地图定位](#)

②加载路径点文件

在菜单栏 [Computing] 右侧的 [waypoint\_maker] 下的 [waypoint\_loader] 加载 waypoints 文件，选择 [Ref] 加载之前保存下来的路点文件，并点击 [Close] 关闭页面。勾选 [waypoint\_loader] ，如下图所示

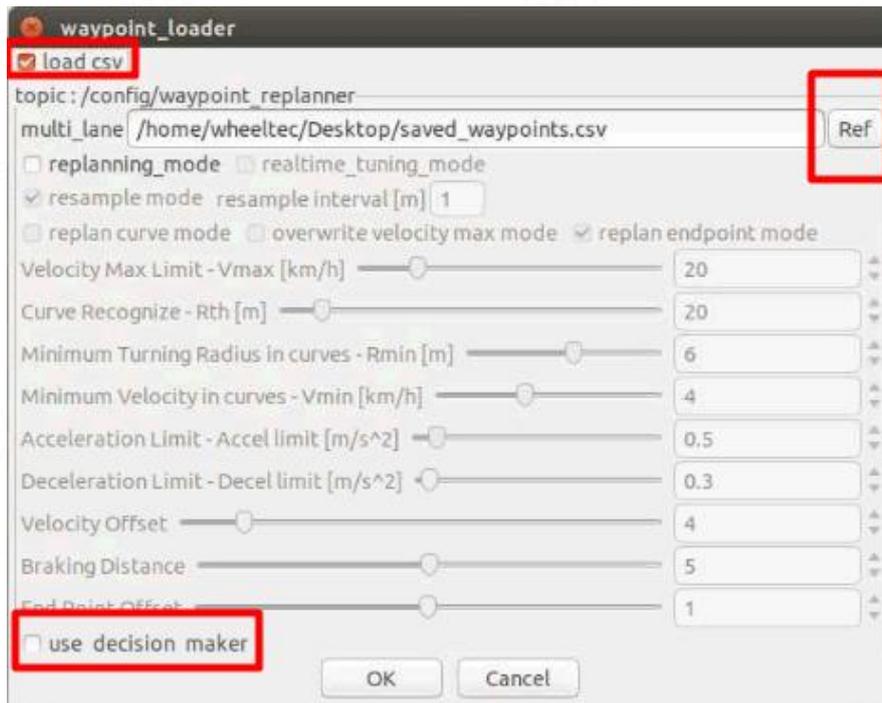


图 2-8-1 加载路径点文件

不勾选[replanning\_mode]时，小车以录制路径时的速度为默认速度进行运动。

勾选[replanning\_mode]时，重新设置路径点的速度，小车直线运动最大速度为[velocity max limit -vmax]，转弯运动最大速度为[minimum velocity in curves]。

建议勾选[replanning\_mode]。

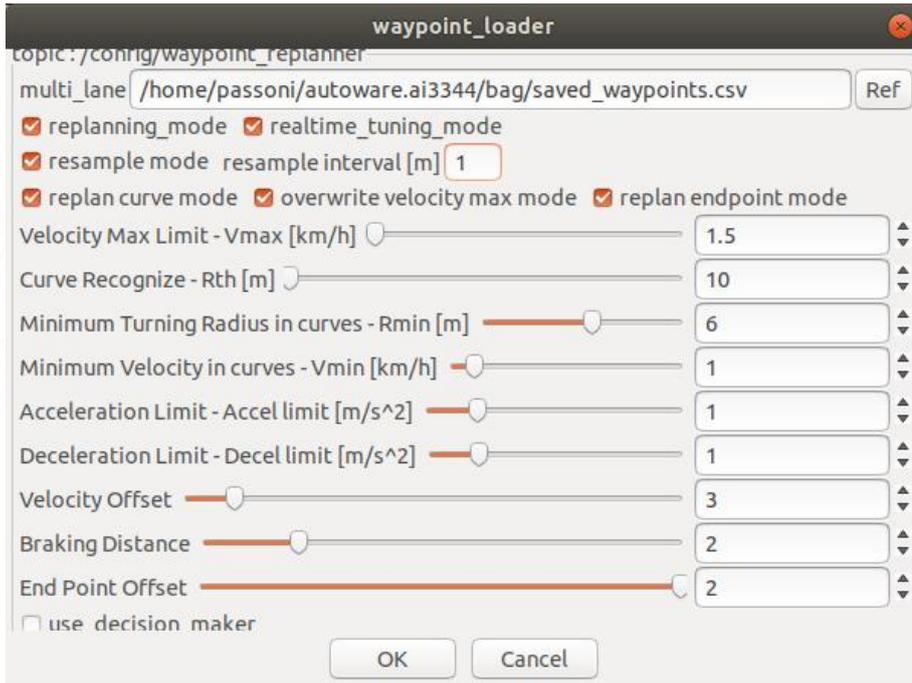


图 2-8-2 重新设定路径点速度

### ③加载代价地图

找到菜单栏 [Computing] 左侧的 [semantics] 下的 [costmap\_generator] , [app]设置好参数之后, 勾选 [costmap\_generator], 相关参数如下图所示。

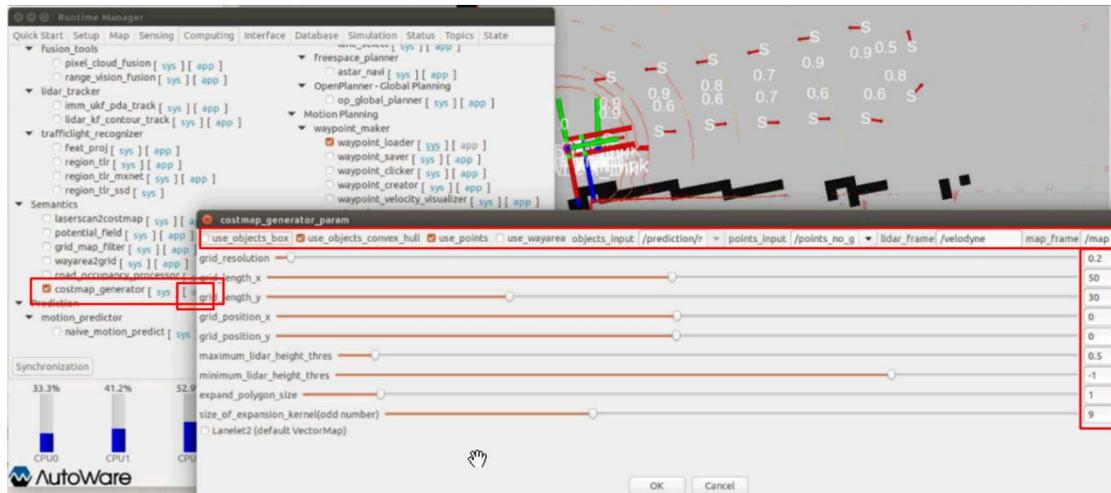


图 2-8-3 加载代价地图

Grid\_length\_x/Grid\_length\_y: 代价地图长、宽

Grid\_position\_x/Grid\_position\_y: 代价地图相对于小车的位置

Maximum\_lidar\_height\_thres (m): 雷达点云最大值 (过滤用)

Minimum\_lidar\_height\_thres (m): 雷达点云最小值 (过滤用)

### ④Lane\_planner

找到菜单栏 [Computing] 右侧的 [lane\_planner], [app]设置好参数之后, 勾选 [lane\_rule]、[lane\_stop]、[lane\_select]。

[lane\_rule]表示设置当红灯亮时, 停止线之前与之后点的个数。具体参数如下图所示:

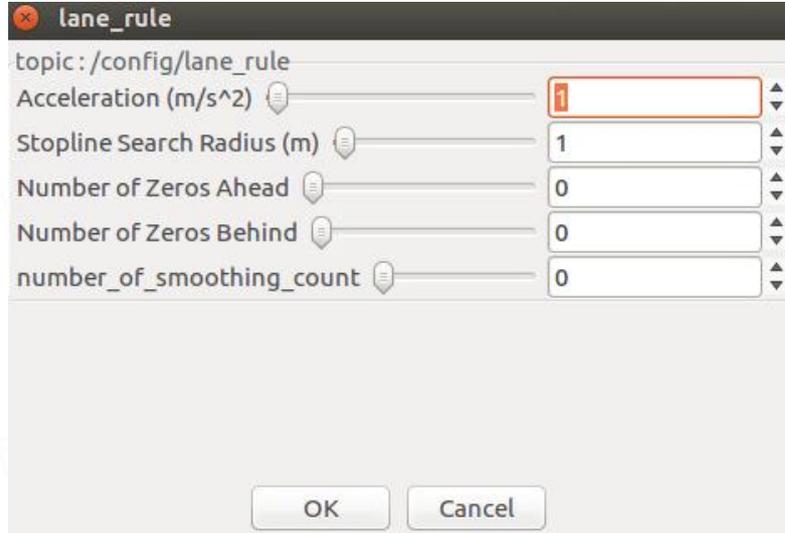


图 2-8-4 lane\_rule 相关参数

**Acceleration:** 表示到达/通过停止线的加速度

**Stopline Search Radius:** 表示搜索停止线圆心, 半径 1m 的 waypoints

**Number of zeros ahead/Number of zeros behind:** 表示停止线前/后的 waypoints 个数, 用于修改停止线前后 waypoints 的速度/加速度

[lane\_stop]用于识别红绿灯等, 根据红绿灯的结果将停止线的 waypoints 变为红色 (停止) 或者绿色 (继续形式)

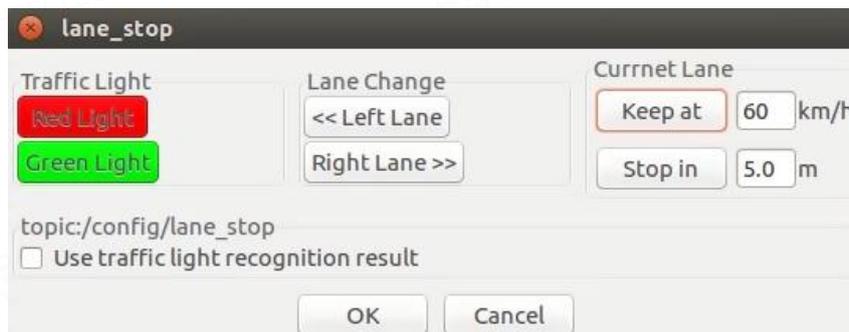


图 2-8-5 lane\_stop 相关参数

[lane\_select]用于将汽车行驶在可以选择的 Lane 上, 如果 [waypoint\_loader] 仅有一条 Lane 可以选择, 那么选择这条路径, 参数默认即可。

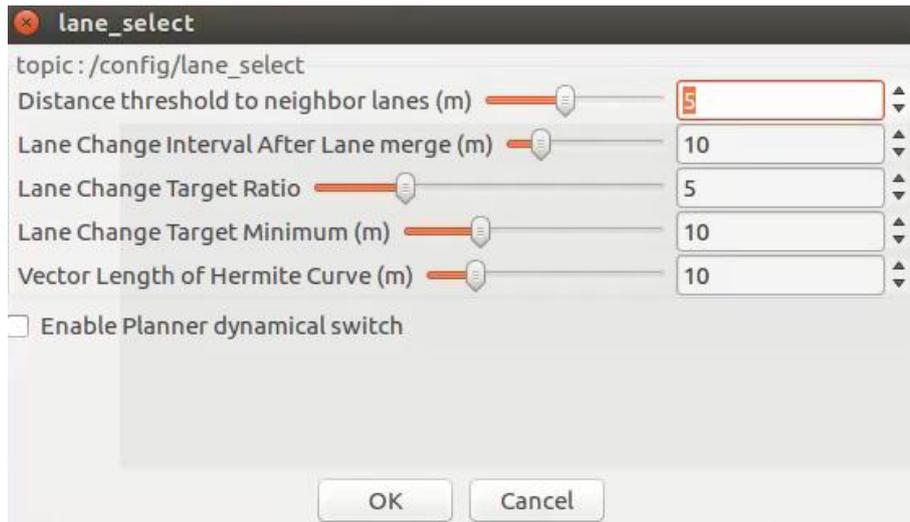


图 2-8-6 lane\_select 相关参数

### ⑤Astar 避障

找到菜单栏 [Computing] 右侧的 [waypoint\_planner]的[astar\_avoid], [app]设置好参数之后, 勾选[astar\_avoid], 相关参数如下图所示

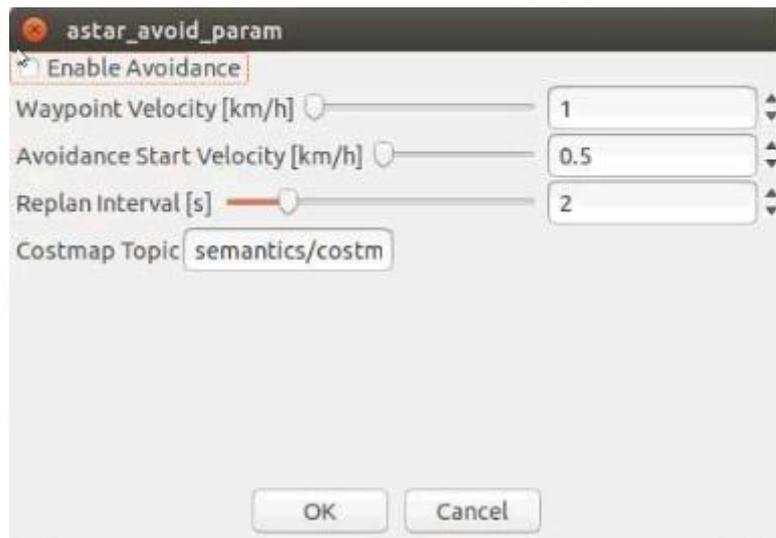


图 2-8-7 astar\_avoid 相关参数

**Enable avoidance:** 是否避障, 取消勾选则遇到障碍物会停下来, 勾选则遇到障碍物会绕开, 勾选的避障效果不稳定, 建议不勾

**Waypoint velocity:** 勾选[astar\_avoid]时, 小车避障时的行驶速度

**Avoidance start velocity:** 小车避障的初始速度

### ⑥避障相关参数设置

找到菜单栏 [Computing] 右侧的 [waypoint\_planner]的[velocity\_set], [app]设置好参数之后, 勾选[velocity\_set], 相关参数如下图所示

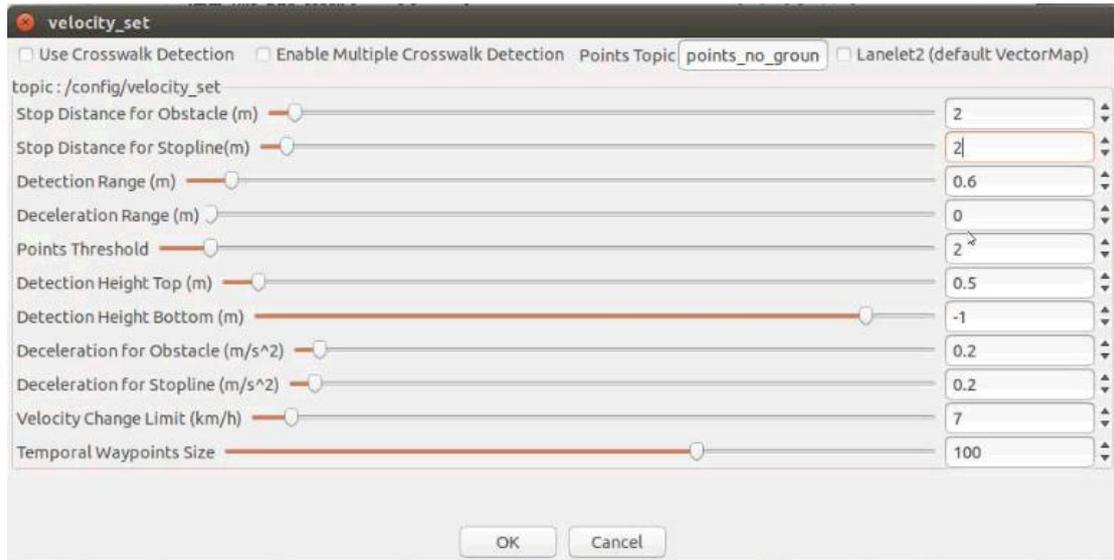


图 2-8-8 velocity\_set 相关参数

**Stop distance for obstacle:** 距离障碍物多远停车

**Stop distance for stopline:** 距离停车线多远停车

**Detection range:** 检测范围，这个是以每个路径点为中心进行检测，检测到范围内存在障碍物，则该路径点设置小车通过速度为 0

**Deceleration range:** 减速范围

**Detection height top/Detection height bottom:** 判断障碍物阈值

#### ⑦速度输出，小车开始导航

打开[Computing]页面，找到 [waypoint\_follower] 下 [pure\_pursuit] 与 [twist\_filter]，设置[app]的一些参数后，勾选 [pure\_pursuit] 与 [twist\_filter] 。

具体参数设置参考[第 2.6 节的⑤步骤内容](#)

## 3.MapTools 工具绘制高精地图

本章只是简单介绍一下高精地图的绘制，满足最基本的导航功能。

### 3.1 环境适配

在 win10 系统下登录 Unity 官网下载相应的版本 <https://unity.cn/releases>，这里最好是 Unity Hub 和 Unity 都在官网下载好相应的包，如下图下载两个安装包文件，然后都需要安装（或者在资料包中找到两个软件，并双击安装）。



图 3-1-1Unity Hub 和 Unity 下载

安装后打开 Unity Hub，在其中点击 installs——locate，找到 Unity 的安装文件夹，点击 Unity——Editor——Unity.exe 文件即可。

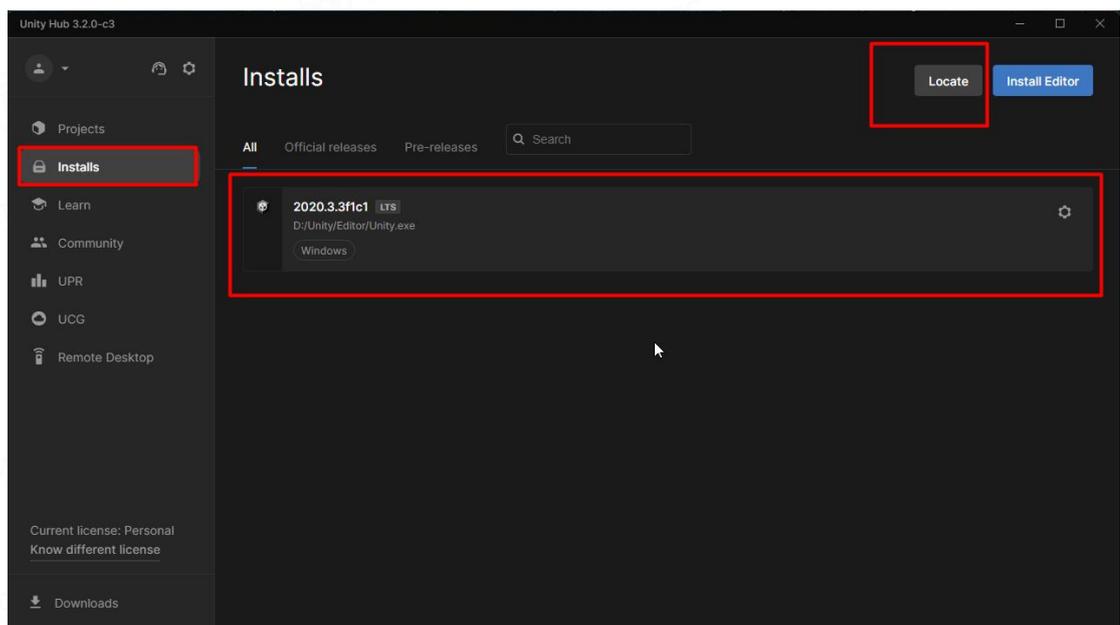


图 3-1-2 Unity Hub 中加载 Unity

接下来在 Unity Hub 里面点击 projects——new project——3D，取消勾选“enable version control ...”选项，然后点击 create project，就进入了 Unity 界面了。

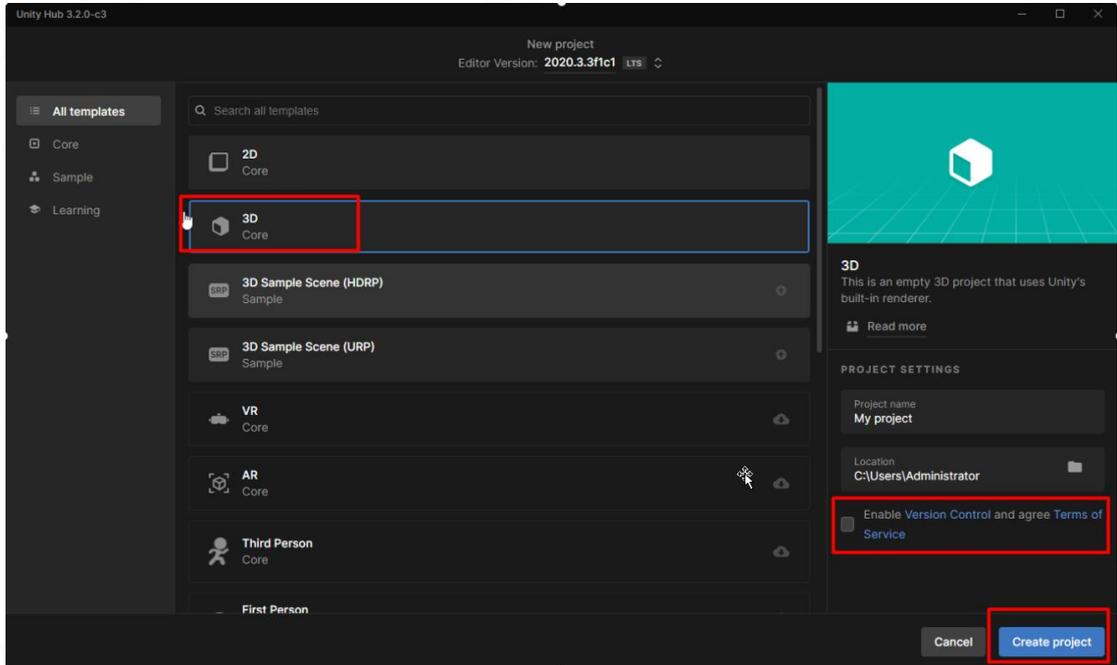


图 3-1-3 Unity Hub 中新建 3D 项目

在 unity 菜单栏中点击 Windows——Package Manager——点击 Package Manager 下面的“+”——Add Package from git URL,填入 com.unity.entities 后点击 add 等待加载下载完成。

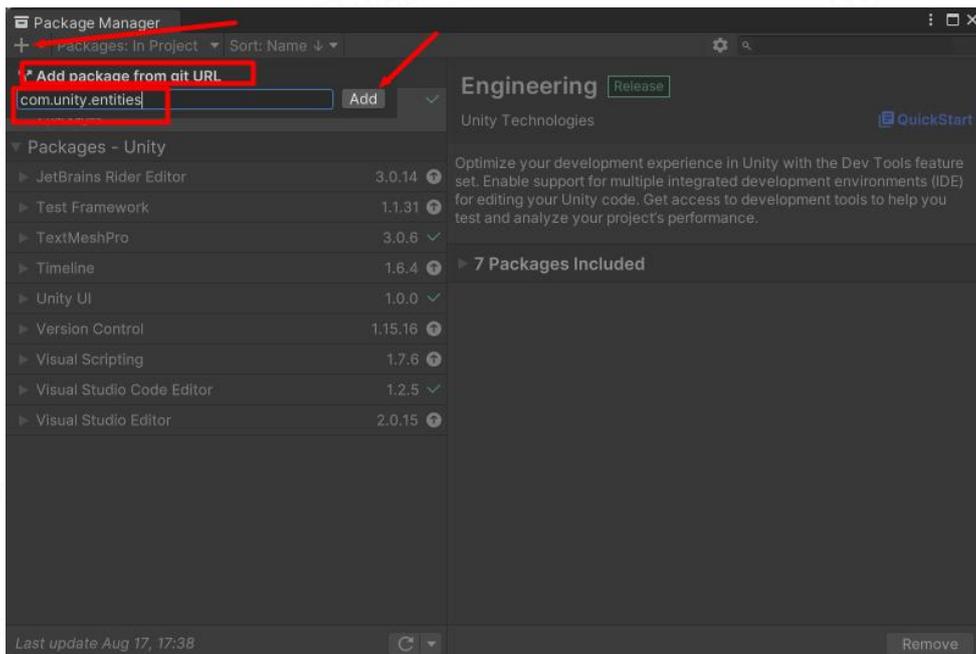


图 3-1-4 添加 com.unity.entities

在资料中找到[MapToolbox-0.1.1.preview.9.zip]压缩包，把它解压在一个全英文的路径下（注：路径中不能有中文以及空格符号）。

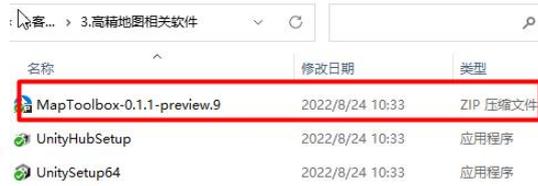


图 3-1-5 Maptoolbox 工具

在 unity 菜单栏中点击 Windows——Package Manager——点击 Package Manager 下面的“+”——Add Package from disk——选择资料中已经解压好的 Maptoolbox 文件夹，其中有一个 json 文件，选中它即可，然后等待一会儿你就会发现在 GameObject 的 UI 下面有一个 Autoware 选项，里面有一个 AutowareADASMap，原本是没有这个的，因为我们导入了 Maptoolbox 工具包的原因，这样我们的制图环境就配置好了。

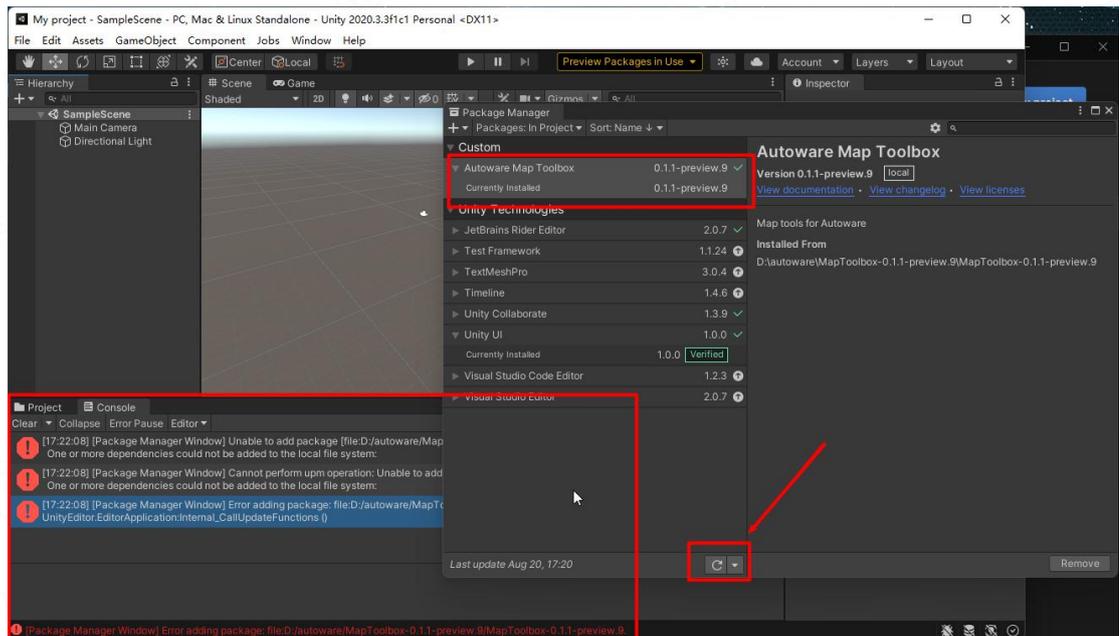


图 3-1-6 Unity 导入 Maptoolbox 工具包

注：假如发生图示中的报错，点击图中刷新按钮的[refresh list]选项，然后重新添加即可。

## 3.2 高精地图简单绘制

### ①加载 pcd 点云地图

把之前建图得到的 pcd 点云地图直接拖拽到 unity 左下角 project 中的 Assets 文件夹当中，然后在把 pcd 点云文件从 Assets 文件夹中拖拽到左侧位置，我们就可以看到点云文件显示在中间位置。

长按鼠标右键可以旋转视角，长按鼠标滑轮可以移动视角，鼠标滑轮可以放大缩小视角。Q 是视角移动，W 是物体移动，E 是旋转。

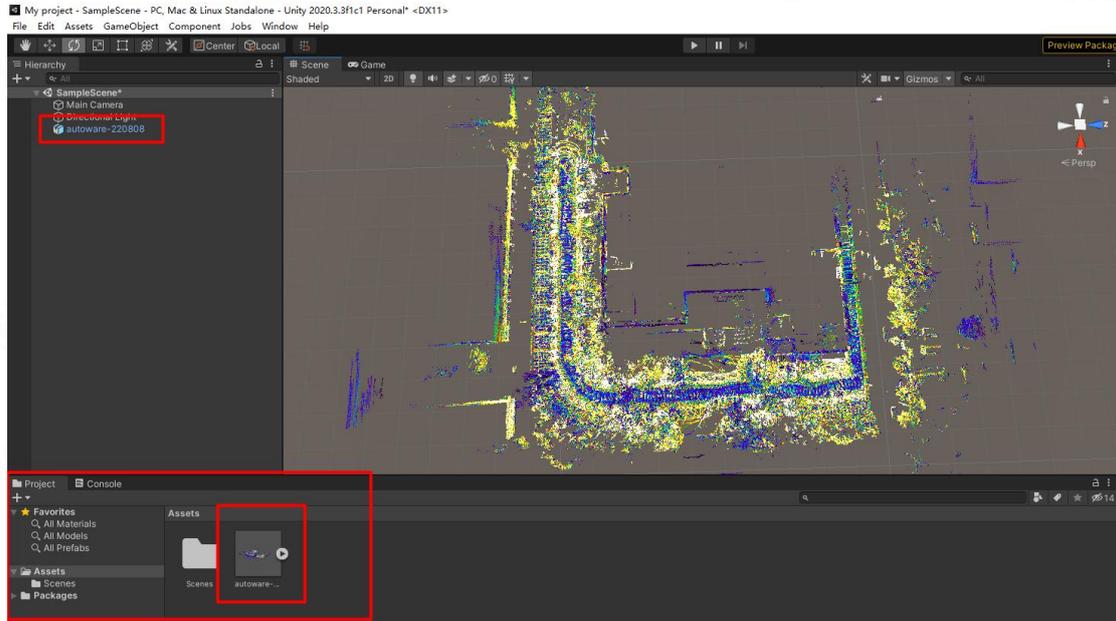


图 3-2-1 加载 pcd 点云地图

## ②固定点云视角

画高精地图时最好固定点云视角，避免点云动，高精地图对不上位置。可以在右上方选择“y”视角，选择“y”视角正方向，然后锁住即可。

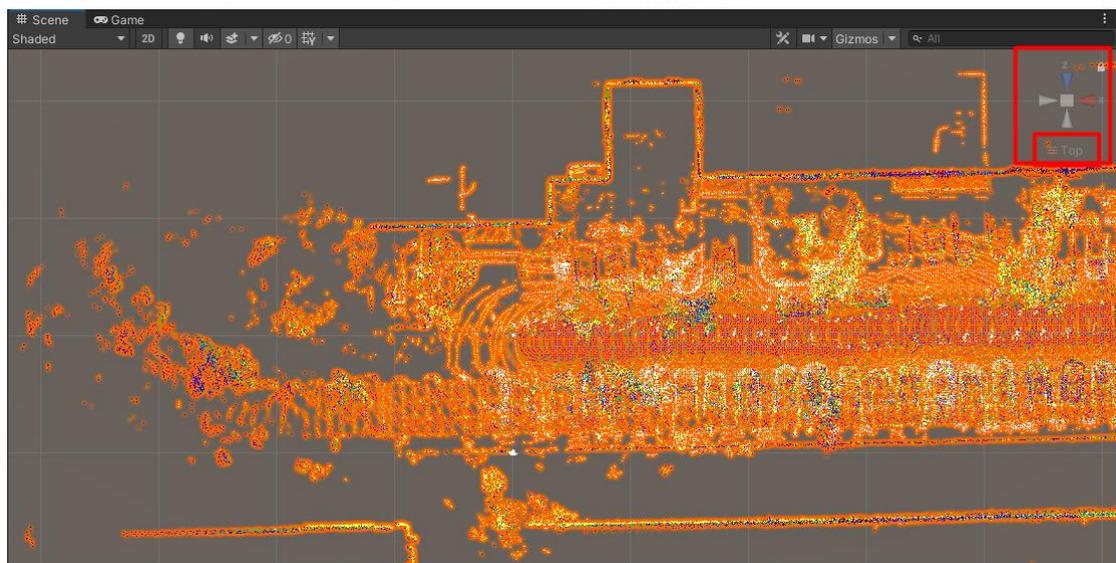


图 3-2-2 固定点云视角

### ③导入 mapprotoolbox 插件

鼠标右击导入 AutowareADASMap，点击它，右侧就会出现一些选项：

Add lane: 添加行驶线，导航就是以行驶线为导航路线，注意行驶线有方向

Add white line: 添加白线，不同行驶线之间以白线相隔

Add road edge: 添加路沿，即道路边缘

Add way area: 车辆可行驶的区域

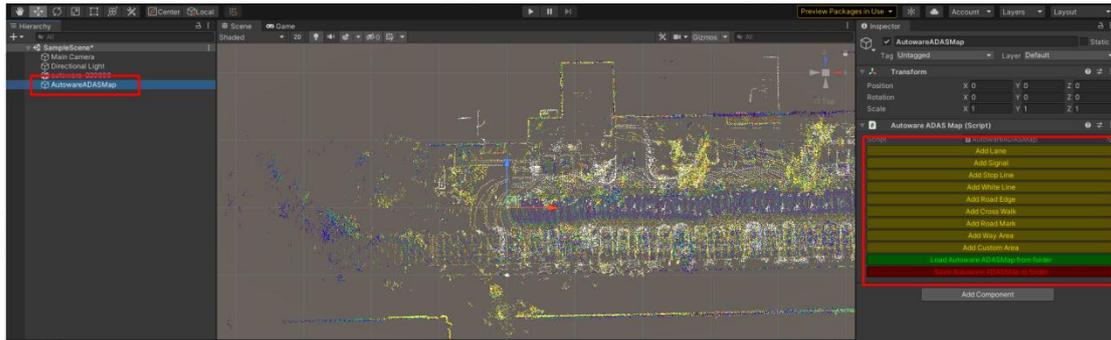


图 3-2-3 导入 mapprotoolbox 插件

### ④绘画直线路段

点击左侧“AutowareADASMap”，然后在点击右侧你想添加的选项，如“Add road edge”，在图中出现一个坐标，拖动坐标即可移动该路线，把它拖到相应位置即可。

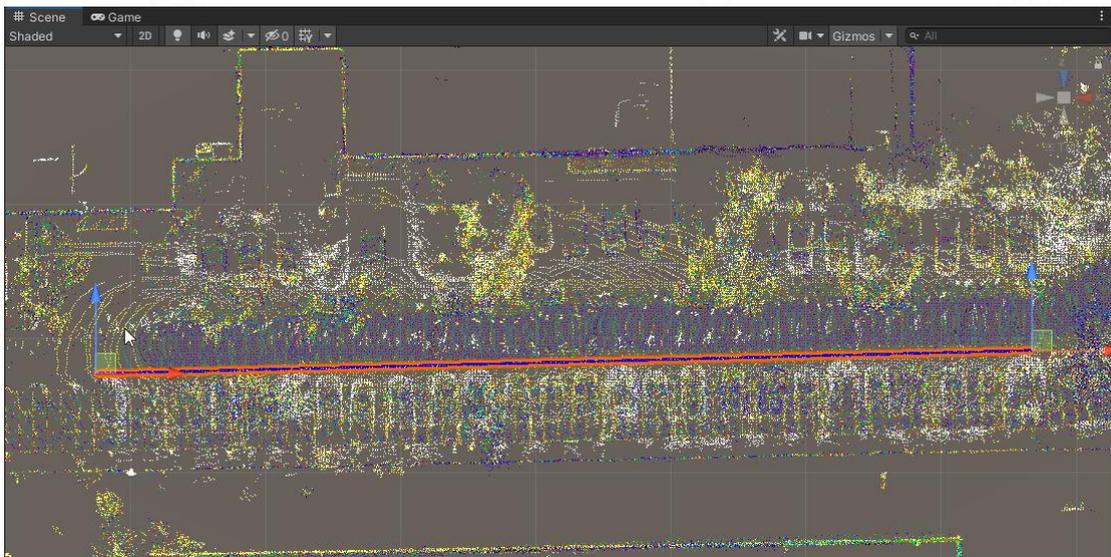


图 3-2-4 绘画直线路段 1

按照以上方法，我们从上到下分别添加了以下几条路线，注意行驶线“add lane”的方向为箭头方向，小车只能沿着箭头方向行驶。

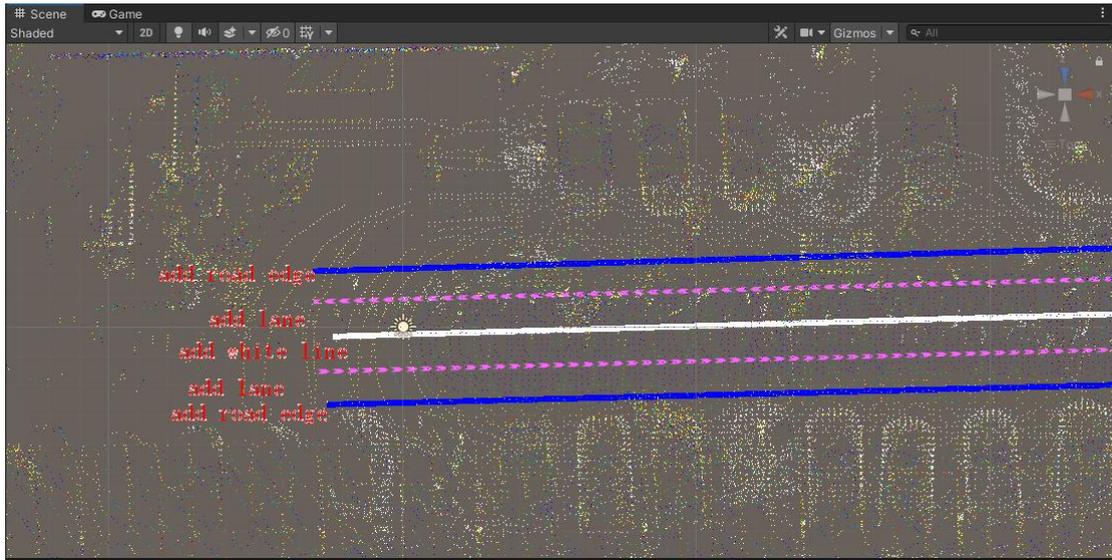


图 3-2-5 绘画直线路段 2

以上几条路线从上到下分别为 road edge——lane——white——lane——road edge。两条道路边缘线 road edge 是需要画的，lane 行驶线是小车规划全局路径的依据，如果小车按照所画的行驶线无法到达目标点，那么小车无法进行全局路径规划，小车无法到达目标点位置，具体 Lane 行驶线按照自己需求画即可，可以只画一条 lane，也可以画多条。

注意画行驶线 lane 时候，最后我们应该选中行驶线的某一条线段，点击右侧的“subdivision”，然后再点击“normal way”，把行驶线分割成一条条小的线段。

注：需要把所有的行驶线分割成小线段，不是行驶线的不用。

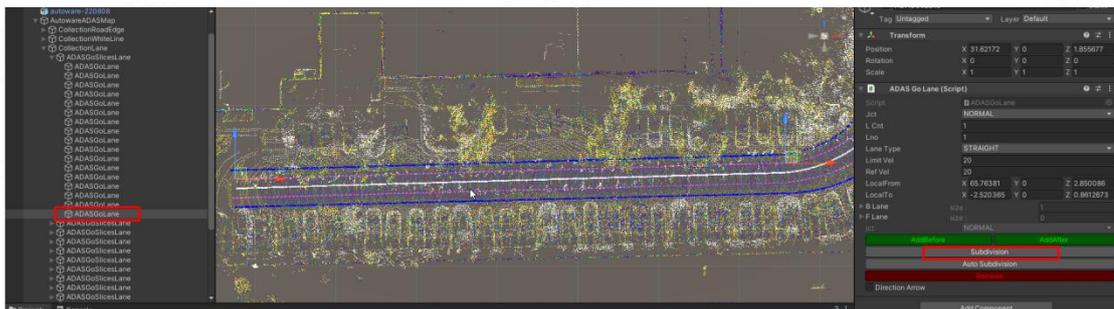


图 3-2-6 绘画直线路段 3

### ⑤绘画转弯路段

在左侧点击选中其中的一小截线段，在右侧点击“addbefore”或者“addafter”即可在该条线段上继续添加。

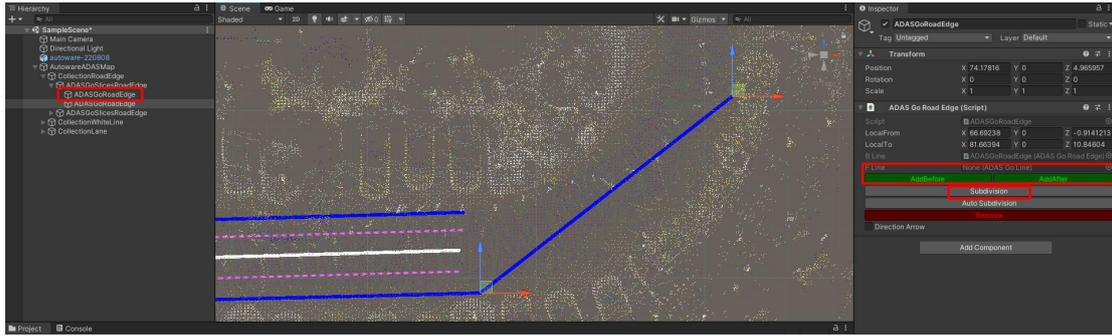


图 3-2-7 绘画转弯路段 1

在转弯等路线不是直线的路段，可以先画一条直线，然后选中该直线，点击“subdivision”，就会在图中出现两个坐标点，拖动两个坐标点即可改变线段形状，拖动到合适位置后，点击右侧的“normal way”，即可保存修改。

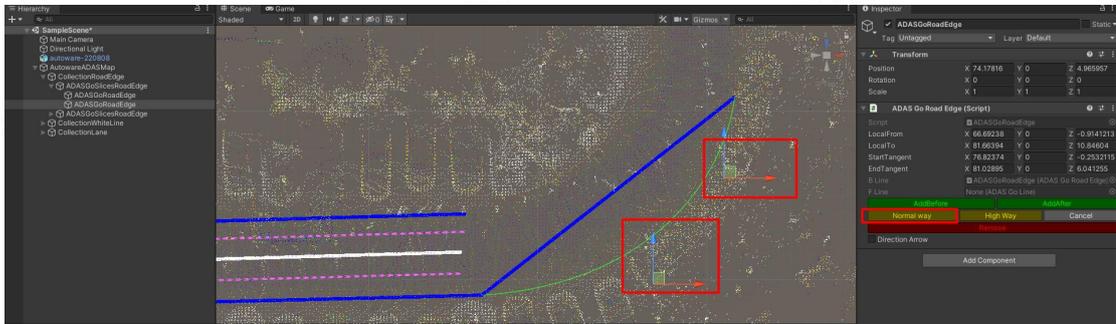


图 3-2-8 绘画转弯路段 2

同样的方法，修改其余几条路线如下：

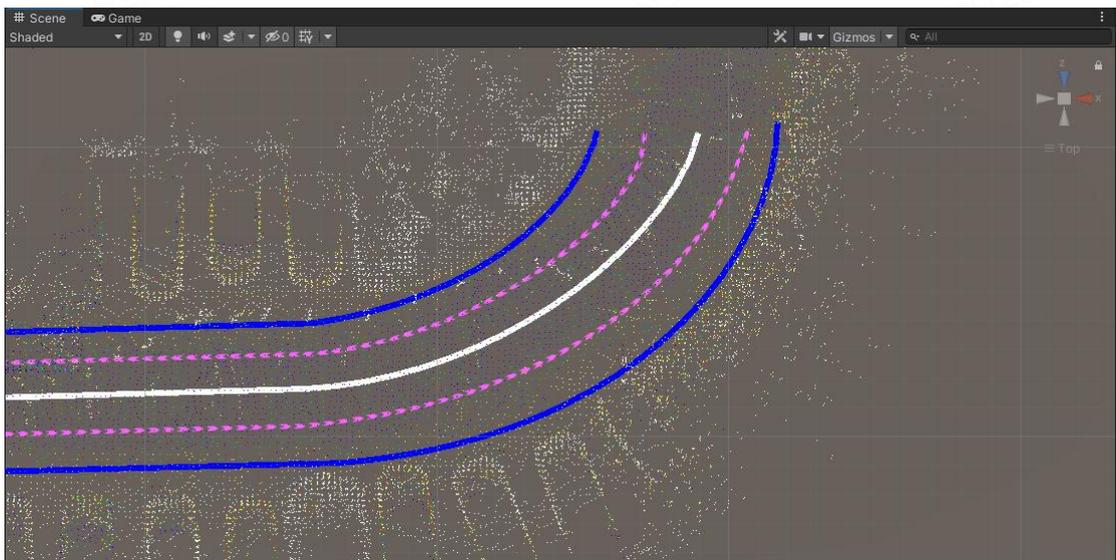


图 3-2-8 绘画转弯路段 3

### ⑥绘画分叉口

首先按照以上方法先把直线道路给画出来，如下所示：

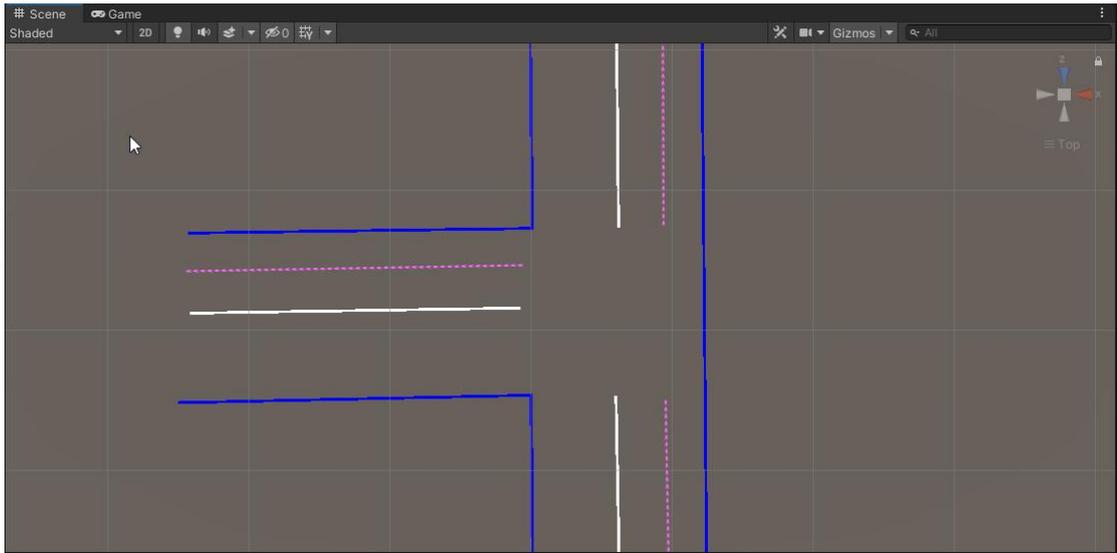


图 3-2-9 绘画交叉口路段 1

然后点击在“AutowareADASMap”下“add lane”添加新的行驶线，连接不同的行驶线使箭头处的行驶线有右转弯，左转弯，或者前进的功能（注意不同行驶线最好连接在同一个点）。

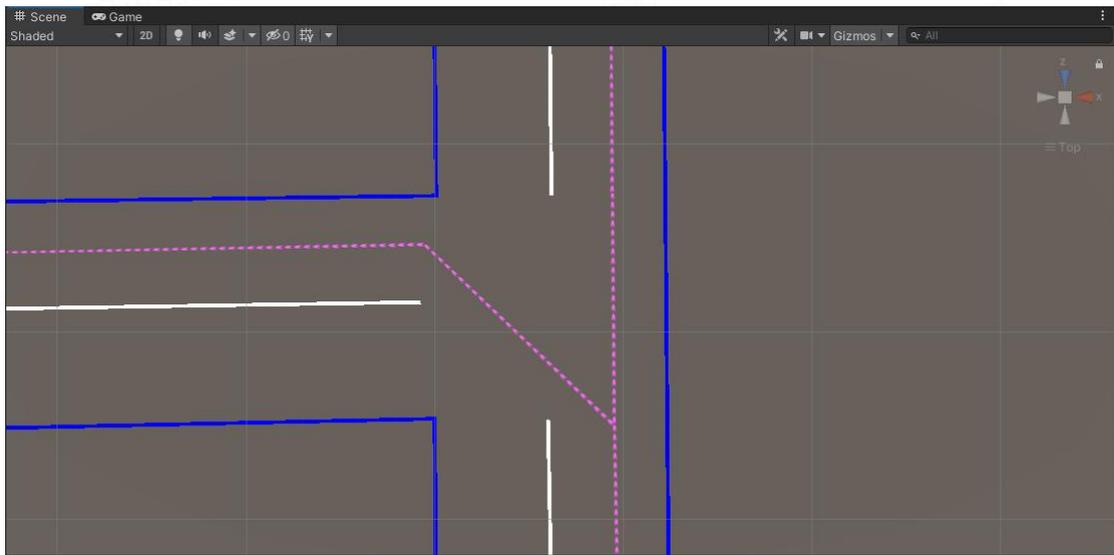


图 3-2-10 绘画交叉口路段 2

然后用以上的方法选中其中一条线段，点击“subdivision”，把直线改成曲线即可。

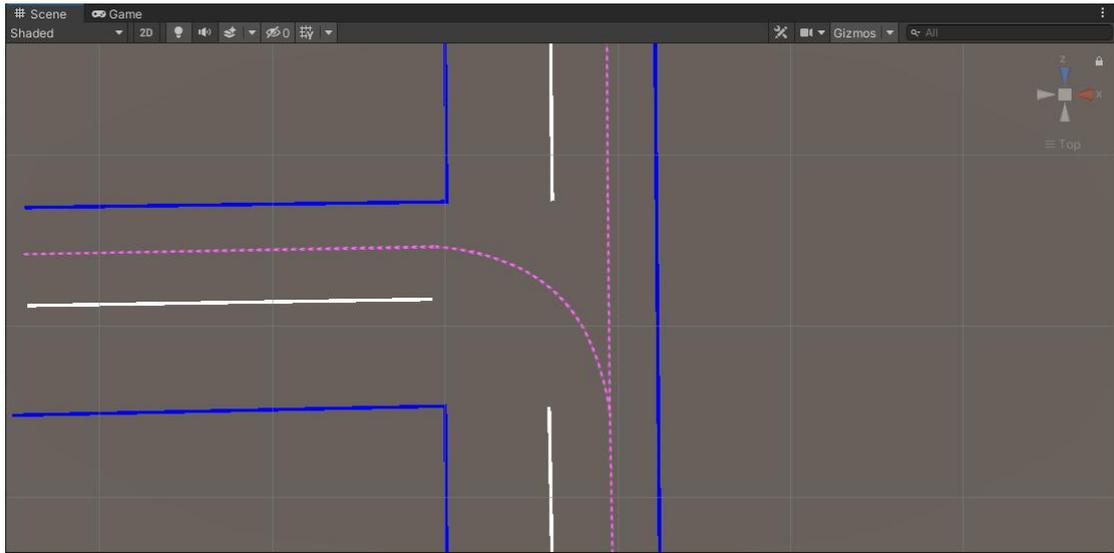


图 3-2-11 绘画交叉口路段 3

### ⑦保存文件

保存时候，需要保存两次算是保存成功。

点击左侧“AutowareADASMap”，然后点击右侧的“save autoware adasmap from folder”，选中需要保存的文件夹路径进行保存。

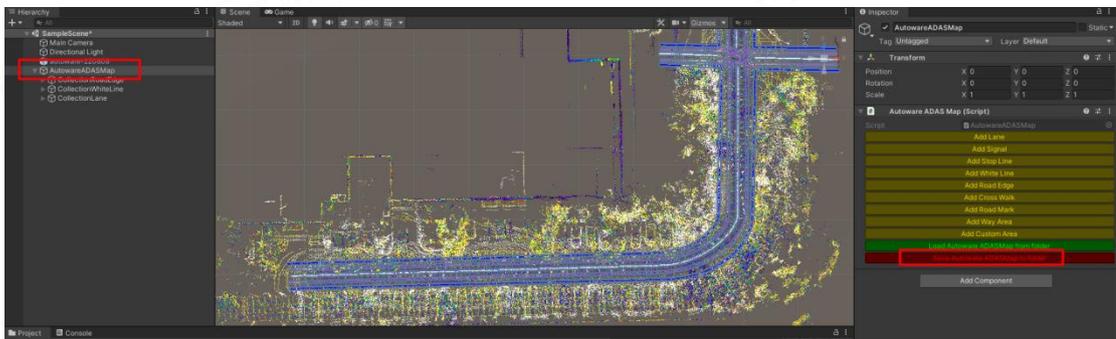


图 3-2-12 保存文件 1

然后我们需要点击“load autoware adasmap from folder”选中刚刚保存的文件夹路径，加载高精地图，这时候，我们发现左侧中，软件已经给不同线段命名排序了，此时点击“save autoware adasmap to folder”保存才是我们需要的。

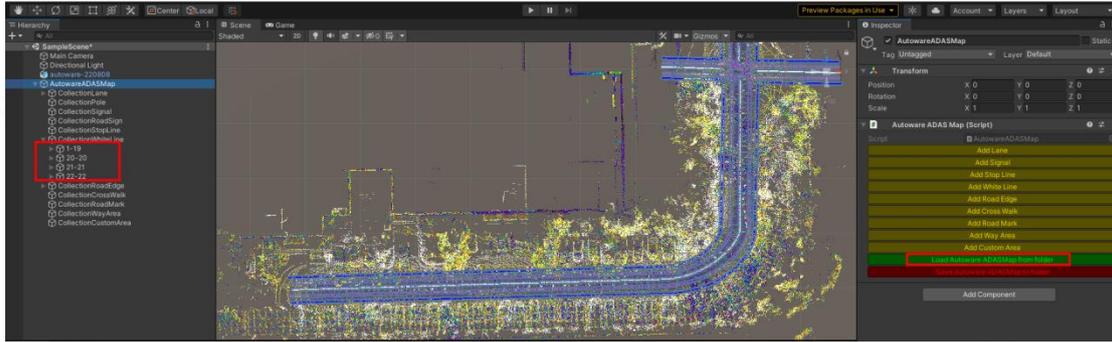


图 3-2-13 保存文件 2

注:

1. lane 所有行驶线需要通过“subdivision”分割成一条条小线段。
2. 绘画跑道等回环型地图时，行驶线最好不要“addbefore”或者“addafter”等方式从头画到尾，可以用“add lane”方式添加新行驶线线段，用新的行驶线线段去衔接。