
千问 Apollo

发行版本 1.0.0

ApolloAuto

2024 年 11 月 02 日

1	基础	3
2	硬件	5
2.1	目前 Apollo 支持的硬件设备有哪些?	5
2.2	如何进行自动驾驶硬件选型?	5
2.3	如何适配新的自动驾驶车辆底盘?	6
3	中间件	7
4	数据	9
4.1	如何转换数据集为 Record 数据包?	9
4.2	如何解压 Record 数据包?	10
4.3	RosBag 和 Record 如何互相转换?	11
4.4	如何将数据包保存至外部挂载的硬盘?	12
5	定位	15
6	感知	17
6.1	Apollo 如何实现 BEV 感知算法?	17
6.2	Apollo 支持哪些感知模型?	17
6.3	如何下载 Apollo 感知模型?	18
7	预测	21
8	导航	23
9	规划	25
9.1	Apollo 如何实现强化学习的轨迹规划?	25
10	控制	27
11	坐标转换	29
11.1	Apollo 的坐标系是如何定义的?	29
12	高精度地图	33
12.1	Apollo 高精度地图格式是什么?	33
12.2	如何将二进制 (bin) 格式的地图转换为文本 (txt) 格式?	34
12.3	如何转换 OpenDRIVE 格式地图到 Apollo 格式地图?	35

13 仿真	37
13.1 Apollo 如何和 Carla 联合仿真?	37
14 Dreamview	39
15 工具	41
16 杂项	43
16.1 如何查看 Apollo 日志?	43
16.2 如何查看 coredump 原因?	44
16.3 如何解决编译依赖下载失败的问题?	45
16.4 拉取 Apollo Docker 失败怎么办?	46

[English](#) | [中文](#) | [🇺🇸🇬🇧🇫🇷](#) | [日本語](#)

我们知道您对Apollo有很多疑问。这里我们总结了 1000 个关于 Apollo 的问题，我们知道这还不够，会逐渐添加答案。希望对您有帮助！

CHAPTER 1

基础

2.1 目前 Apollo 支持的硬件设备有哪些？

目前 Apollo 支持的硬件设备有哪些？例如车辆底盘、计算单元、传感器等。

- 维护者: daohu527@gmail.com
- 版本: 1.0.0
- 日期: 05/09/2024
- 描述:

2.1.1 回答

Apollo 对多种自动驾驶硬件设备提供了全面支持，以方便客户进行自动驾驶硬件选型，详细的硬件支持列表，请参考[官方硬件开发平台](#)页面。

如果您是硬件设备厂商，也欢迎你加入 Apollo 硬件生态，共同推动自动驾驶技术的发展。

2.2 如何进行自动驾驶硬件选型？

- 维护者: daohu527@gmail.com
- 版本: 1.0.0
- 日期: 05/19/2024
- 描述:

2.2.1 回答

2.3 如何适配新的自动驾驶车辆底盘？

我想适配一辆新的自动驾驶车辆，但是不在当前 Apollo 支持的车辆列表里面，我需要做哪些工作？

- 维护者: daohu527@gmail.com
- 版本: 1.0.0
- 日期: 24/07/2024
- 描述:

2.3.1 回答

要适配新的自动驾驶车辆，首先需要确保车辆满足线控要求，即具备通过电信号来控制车辆的能力，包括：线控油门、线控刹车、线控转向和线控档位。其次，这些线控系统的性能必须满足自动驾驶控制的要求，具体标准因不同场景而异，例如卡车和乘用车的标准就有所不同。最后，还需要获取车辆的 DBC 控制协议，以便根据协议发送控制命令。

接下来我们详细介绍这几个部分。

线控

传统车辆是通过油门、刹车踏板、方向盘和档位杆来控制行驶，而自动驾驶车辆则是通过 canbus 总线发送命令来实现控制。因此，车辆必须支持通过电信号来控制，即具备线控油门、线控刹车、线控转向和线控档位功能。

线控性能

车辆不仅要支持线控，其线控性能也必须达到一定标准。例如，从接收控制指令到响应的的时间，以及车辆的转向精度等。线控性能越好，车辆控制就越方便；如果线控性能差，控制难度就会增加，甚至可能无法支持自动驾驶。详细的线控要求可以参考 [Apollo 线控标准](#)。

注意：这是乘用车的底盘要求，商用车的要求可能有所不同。为了支持不同的底盘和动力学特性，控制模块需要进行相应的调整。

底盘协议

即使车辆支持线控且性能满足要求，由于各厂家控制协议不同，仍需适配车辆的底盘控制协议，才能实现车辆控制。为了简化适配过程，我们提供了 [adbtool](#) 工具，它可以将车辆的 DBC 协议转换为 Apollo 中的 C++ 车辆控制协议。这样，Apollo 自动驾驶系统的控制命令就能转换为车辆的 canbus 命令，从而控制车辆行驶。

以上就是在 Apollo 中新增加车辆底盘的过程。

CHAPTER 3

中间件

4.1 如何转换数据集为 Record 数据包？

为了评估 Apollo 感知算法在特定数据集上的性能，我希望将数据集中的数据转换成 Apollo 所需的数据包，以便在 Apollo 平台上进行测试。

- 维护者: daohu527@gmail.com
- 版本: 1.0.0
- 日期: 06/13/2024
- 描述:

4.1.1 回答

可以使用 `adataset` 工具进行数据集到 `record` 包的转换。目前已经支持的数据集有: `apolloscape`、`KITTI`、`nuScenes`。

以 `nuScenes` 数据集为例。设置 `dataset_path` 为 `nuScenes` 数据集所在路径。默认情况下，数据包输出到当前目录。

```
adataset -d=n -i=dataset_path
```

`nuScenes` `record` 包的名称是 `scene_token.record`，`KITTI` 是 `result.record`，而 `ApolloScape` 是 `frame_id.record`。

4.2 如何解压 Record 数据包？

如何解压 Record 数据包，提取其中的点云、图像等消息？

- 维护者: daohu527@gmail.com
- 版本: 1.0.0
- 日期: 05/09/2024
- 描述:

4.2.1 回答

自动驾驶系统实时录制了自动驾驶各个模块发布的消息，并且保存在 Record 数据包中，用户可以从数据包中提取数据进行模型训练、问题分析等。

那么如何从数据包中解压这些数据呢？

安装

cyber_record 是一款纯 python 实现的 Record 读写工具，同时提供命令行和 API2 种方式，好处是轻量化，跨平台，缺点是速度较慢。

```
pip3 install cyber_record record_msg
```

读取图像

指定图像消息的 topic，和需要保存的路径 output_path，运行以下代码。

```
from cyber_record.record import Record
from record_msg.parser import ImageParser

image_parser = ImageParser(output_path='../test')
for topic, message, t in record.read_messages():
    if topic == "/apollo/sensor/camera/front_6mm/image":
        image_parser.parse(message)
        # or use timestamp as image file name
        # image_parser.parse(image, t)
```

读取点云

指定点云消息的 topic，和需要保存的路径 output_path，运行以下代码。

```
from cyber_record.record import Record
from record_msg.parser import PointCloudParser

pointcloud_parser = PointCloudParser('../test')
for topic, message, t in record.read_messages():
    if topic == "/apollo/sensor/lidar32/compensator/PointCloud2":
        pointcloud_parser.parse(message)
        # other modes, default is 'ascii'
```

(续下页)

(接上页)

```
# pointcloud_parser.parse(message, mode='binary')
# pointcloud_parser.parse(message, mode='binary_compressed')
```

如果需要加速读取过程，可以使用过滤读取，过滤读支持指定 topic 和时间过滤，会加速消息读取处理过程。

```
def read_filter_by_both():
    record = Record(file_name)
    for topic, message, t in record.read_messages('/apollo/canbus/chassis', \
        start_time=1627031535164278940, end_time=1627031535215164773):
        print("{} {}, {}".format(topic, type(message), t))
```

4.3 RosBag 和 Record 如何互相转换？

如何有效地将 ROS Bag 转换为 Apollo Record，以及将 Apollo Record 转回 ROS Bag？

- 维护者: daohu527@gmail.com
- 版本: 1.0.0
- 日期: 05/14/2024
- 描述:

4.3.1 回答

ROS Bag 和 Apollo Record 是两种不同的数据保存格式，由于这两种格式在文件结构和消息定义上存在差异，因此需要进行数据转换。

转换过程可归纳为以下三个关键步骤：

1. **读取 ROS Bag 数据**：首先，我们需要使用相应的工具或库打开 ROS Bag 文件，并从中读取存储的 ROS 消息。在这一步中，我们将获取 ROS 消息的类型，为后续转换做准备。
2. **转换消息格式**：在读取了 ROS 消息后，我们需要将这些消息从 ROS 格式转换为 Apollo 系统所支持的格式。这通常涉及到解析 ROS 消息的内容，并根据 Apollo 的消息定义重新组织这些数据。
3. **保存为 Apollo Record**：完成消息格式的转换后，我们将新的 Apollo 消息保存为 Apollo Record 格式的文件。这样，原本在 ROS 系统中使用的数据就可以在 Apollo 系统中被读取和使用了。

为了简化上述转换过程，我们提供了 `bag_convert` 工具。使用此工具，用户只需指定 ROS Bag 文件的路径和输出 Apollo Record 文件的路径，即可轻松完成数据格式的转换。

安装

```
pip3 install bag_convert
```

Ros Bag 转 Cyber Record

```
bag_convert -m=b2r -b=data/input.bag -r=data/output.record
```

Cyber Record 转 Ros Bag

```
bag_convert -m=r2b -r=data/input.record -b=data/output.bag
```

4.4 如何将数据包保存至外部挂载的硬盘？

由于自动驾驶每天录制的数据包比较大，计划使用外部硬盘来保存这些数据。请问如何实现？

- 维护者：daohu527@gmail.com
- 版本：1.0.0
- 日期：11/02/2024
- 描述：

4.4.1 回答

默认情况下，数据会存储在容器中，但为了实现数据持久化并防止数据丢失，需要将外部硬盘挂载到主机的特定目录，再将该目录挂载到容器内。这样，数据将保存在硬盘上，不会随容器停止而丢失。

4.4.2 在 Ubuntu 下挂载硬盘到容器的步骤

1. 识别外部硬盘设备

```
lsblk
```

找到你的外部硬盘设备名（例如 `/dev/sdb1`）。

2. 创建挂载目录

创建一个用于挂载硬盘的目录（例如 `/apollo/data/bag`）。

```
cd apollo
sudo mkdir -p data/bag
```

3. 挂载硬盘

使用 `mount` 命令将硬盘挂载到该目录。

```
sudo mount /dev/sdb1 data/bag
```

4. 【跳过】启动容器并挂载目录

在启动容器时，将该挂载目录绑定到容器内部的路径（例如 `/data`）。

```
docker run -v /mnt/external_drive:/data your_container_image
```

> 因为 `apollo` 目录已经默认挂载到容器，因此可以跳过该步骤。

这样，外部硬盘的数据就可以通过容器内的 */data* 目录访问，并在容器重启或停止后仍然保留在硬盘中。

4.4.3 录制数据包

录制数据包，并且指定保存路径。

```
cyber_recorder record -a -o=data/bag
```


CHAPTER 5

定位

6.1 Apollo 如何实现 BEV 感知算法？

如何运行 Apollo 中的 BEV 感知算法？以及如何对现有算法进行改进？

- 维护者: daohu527@gmail.com
- 版本: 1.0.0
- 日期: 05/19/2024
- 描述:

6.1.1 回答

6.2 Apollo 支持哪些感知模型？

Apollo 支持哪些感知任务，以及对应的感知模型有哪些？

- 维护者: [<daohu527@gmail.com>](mailto:daohu527@gmail.com)
- 版本: 1.0.0
- 日期: 10/30/2024
- 描述:

6.2.1 回答

Apollo 支持的感知任务主要包括：目标检测、目标跟踪、红绿灯检测、车道线识别以及轨迹预测。这些任务均采用最新的深度学习算法。由于每年都会更新模型，以下是对这些任务及其对应的深度学习模型的整理。

我们已在 [GitHub](#) 上注明了这些模型的用途和下载地址，Apollo 支持的模型详情请参考 [链接](#)。

6.3 如何下载 Apollo 感知模型？

如何获取 Apollo 预训练的感知模型，并查看系统中已安装有哪些模型？

- 维护者：<daohu527@gmail.com>
- 版本：1.0.0
- 日期：05/19/2024
- 描述：

6.3.1 回答

为验证自动驾驶算法，Apollo 支持多种任务和模型。由于模型文件较大，自 8.0 版本起，我们已将模型从源码中移除，并采用 *amodel* 工具进行模型管理，包括安装、查看和卸载模型。

安装 amodel

在容器中，使用以下命令安装 amodel。

```
sudo pip3 install modules/tools/amodel
```

备注

注意：由于 amodel 1.0 与 2.0 不兼容，我们将手动安装至容器中，作为临时解决方案。

下载模型

使用模型名称或地址链接安装模型。模型下载地址 [链接](#)。

```
// 示例
amodel install center_point_paddle
```

查看模型

查看系统中已安装的模型。

```
amodel list
```


CHAPTER 7

预测

CHAPTER 8

导航

9.1 Apollo 如何实现强化学习的轨迹规划？

Apollo 中实现了通过强化学习进行车辆轨迹规划，如何运行？以及如何对现有强化学习算法进行改进？

- 维护者: daohu527@gmail.com
- 版本: 1.0.0
- 日期: 05/19/2024
- 描述:

9.1.1 回答

CHAPTER 10

控制

11.1 Apollo 的坐标系是如何定义的？

Apollo 中的坐标系有哪些，以及它们是如何定义的？

- 维护者: daohu527@gmail.com
- 版本: 1.0.0
- 日期: 05/09/2024
- 描述:

11.1.1 回答

在 Apollo 项目中，涉及到了多种坐标系，每种坐标系都有其特定的定义和应用场景。以下是 Apollo 中常见的坐标系及其定义：

- 全球地理坐标系
- 局部坐标系
- 车辆坐标系

全球地理坐标系

Apollo 采用 **WGS84 (World Geodetic System 1984)** 作为标准坐标系来表示物体的经度和纬度。WGS84 是一种以地球椭圆体为参考的坐标系，它使用经度和纬度两个角度值来唯一确定地球表面上除北极点之外的所有点。其中，经度表示东西方向的位置，取值范围为 -180° 至 180° ；纬度表示南北方向的位置，取值范围为 -90° 至 90° 。

WGS84 坐标系广泛应用于地理信息系统 (GIS) 服务，例如地图绘制、定位、导航等。

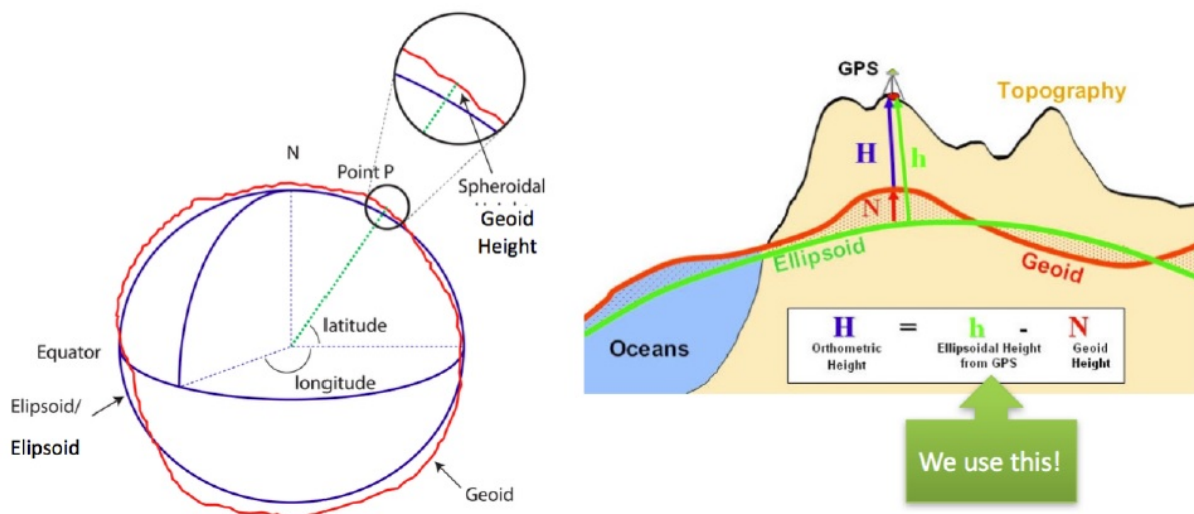


图 1: world_geodetic_system

局部坐标系

局部坐标系是一种以地球表面某一点为原点的三维笛卡尔坐标系，其 X 轴指向东，Y 轴指向北，Z 轴指向天。它常用于表示车辆或传感器等移动物体的相对位置。

墨卡托正形投影 (Universal Transverse Mercator, UTM) 是一种将地球表面划分为 60 个区域的二维投影坐标系。每个区域称为一个投影带，宽度为 6 度经度，并使用横轴墨卡托投影进行投影。

在 Apollo 系统中，UTM 坐标系统在定位、Planning 等模块中作为局部坐标系使用。

车辆坐标系

以上就是在 Apollo 项目中常见的坐标系及其定义。这些坐标系在自动驾驶系统的不同模块中发挥着重要作用，如地图定位、车辆控制、环境感知等。通过合理地使用这些坐标系，可以有效地实现自动驾驶系统中的各种功能。

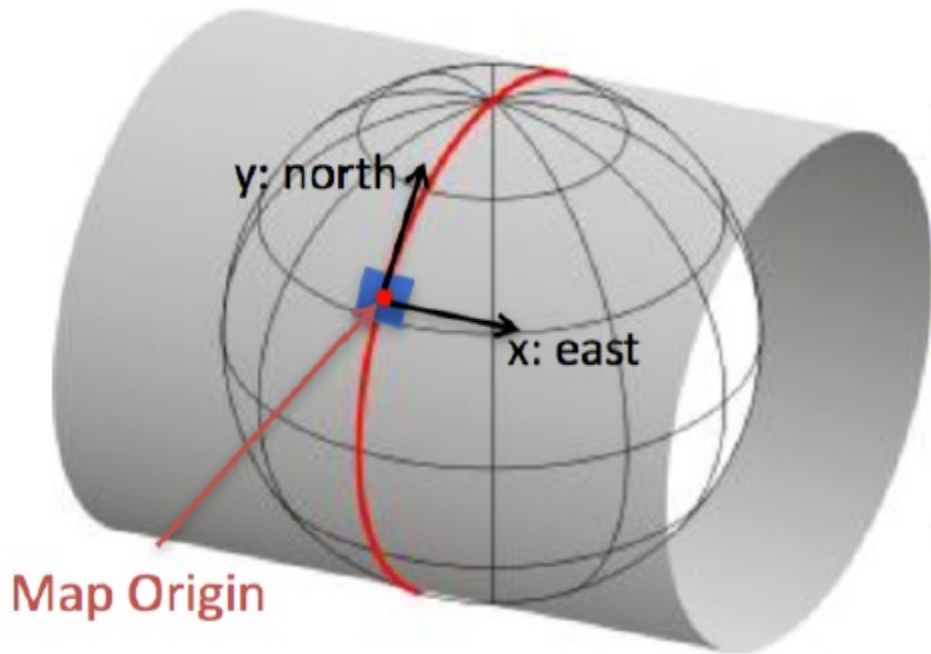


图 2: local_coordinate_system

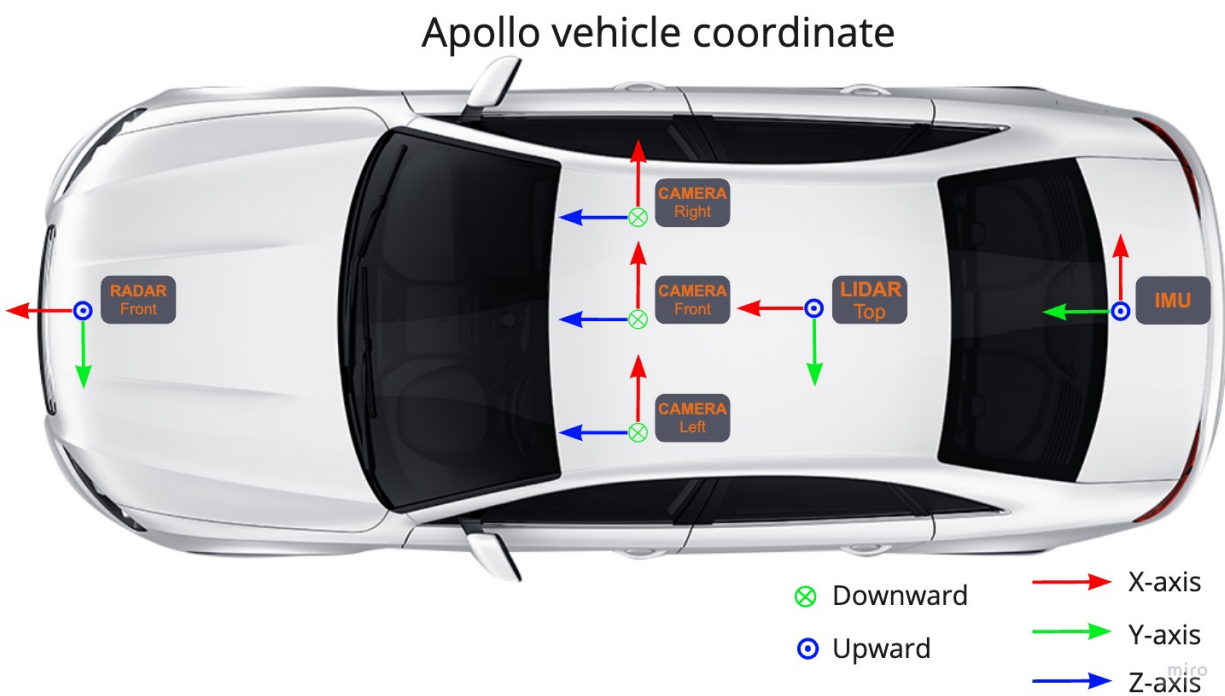


图 3: vehicle_coordinate_system

12.1 Apollo 高精度地图格式是什么？

Apollo 高精度地图的格式是什么？请简要描述其特点和应用场景。

- 维护者：daohu527@gmail.com
- 版本：1.0.0
- 日期：05/09/2024
- 描述：

12.1.1 回答

Apollo 高精度地图格式参考了ASAM OpenDRIVE，针对自动驾驶的实际需求进行了优化。详细的地图格式请参考Apollo 高精度地图格式规范。

Apollo 的优化主要体现在两个方面：车道的离散化表示和车道元素的简化。

连续到离散

OpenDRIVE 标准中，车道线分为以下几种类型：直线、螺旋线、弧线、参数三次曲线、三次多项式（已弃用）。

以直线为例，通过起点坐标 (x, y)、角度 hdg 和长度来定义，从而可以计算出直线上任意一点的坐标。

```
<planView>
  <geometry s="0.0000000000000000e+00"
    x="-4.7170752711170401e+01"
    y="7.2847983820912710e-01"
    hdg="6.5477882613167993e-01"
    length="5.7280000000000000e+01">
  </line/>
```

(续下页)

(接上页)

```
</geometry>
</planView>
```

然而，在自动驾驶系统中，规划算法通常基于采样后的点进行工作。对于连续的曲线，这意味着需要实时进行计算，增加了系统的计算负担。为了简化这一过程，Apollo 选择将车道离散化，将连续的曲线转换为一系列离散的点，这些点可以直接用于自动驾驶的规划和决策过程。

简化车道元素

在 OpenDRIVE 中，只提供了车道中心线的坐标，但车道的左右边界则需要根据车道宽度和偏移量进行计算得出。这种设计增加了地图使用的复杂性，特别是在判断自动驾驶车辆是否在车道内时，需要额外计算车道的左右边界。

为了简化这一过程，Apollo 在地图中直接保存了每条车道的左右边界和邻接关系。这意味着，在判断自动驾驶车辆位置时，无需再进行实时计算，可以直接使用地图中存储的边界信息。这一设计显著提高了地图的易用性和效率。

基于以上 2 点，Apollo 的高精度地图通过离散化车道和简化车道元素，为自动驾驶系统提供了更加高效、易用的地图数据支持。

12.2 如何将二进制（bin）格式的地图转换为文本（txt）格式？

二进制格式地图和文本格式地图的区别是什么？它们之间如何相互转换？

- 维护者：daohu527@gmail.com
- 版本：1.0.0
- 日期：05/09/2024
- 描述：

12.2.1 回答

Apollo 高精度地图主要基于 protobuf 协议进行实现，其存储方式支持两种文件格式：bin 和 txt。

- **bin 格式**是一种紧凑型的文件格式，其优势在于能够显著减少文件的大小，从而减少存储和传输的开销。然而，由于它是二进制编码，因此无法直接通过文本编辑器进行阅读。
- **txt 格式**是基于文本的存储方式，使得地图数据可以直观地以文本形式展现，便于开发人员查看和调试。然而，这种格式会占用较大的存储空间，并且传输效率相对较低。

在实际应用中，为了确保地图数据的高效存储和传输，Apollo 系统通常选择使用 bin 格式进行地图的保存和加载。而 txt 格式则主要作为调试和查看的辅助手段，方便开发人员理解和验证地图数据。

二进制 (bin) 转文本 (txt)

1. 安装 protoc

安装 protoc, 参考[Installing protoc](#)。

2. 运行命令

```
# modules/map/data/borregas_ave/base_map.bin (替换为你想要转换的地图)

protoc --decode apollo.hdmap.Map modules/map/proto/map.proto < modules/map/data/
↪borregas_ave/base_map.bin > base_map.txt
```

3. 转换后的地图文件 base_map.txt 保存在当前工作目录下。

文本 (txt) 转二进制 (bin)

1. 安装 protoc

同上。

2. 运行命令

```
# modules/map/data/demo/base_map.txt (替换为你想要转换的地图)

protoc --encode apollo.hdmap.Map modules/map/proto/map.proto < modules/map/data/demo/
↪base_map.txt > base_map.bin
```

3. 转换后的地图文件 base_map.bin 保存在当前工作目录下。

12.3 如何转换 OpenDRIVE 格式地图到 Apollo 格式地图?

如何将这 2 种地图格式进行相互转换?

- 维护者: daohu527@gmail.com
- 版本: 1.0.0
- 日期: 05/09/2024
- 描述:

12.3.1 回答

以下是 2 种地图格式的转换方法。

OpenDRIVE 转 Apollo

可以使用imap工具来转换 OpenDRIVE 格式地图到 Apollo 格式地图。

1. 安装 imap

```
pip3 install imap_box
```

2. 转换格式

```
# data/town.xodr (OpenDRIVE格式地图)
# data/apollo_map.txt (Apollo格式地图)
imap -f -i data/town.xodr -o data/apollo_map.txt
```

Apollo 转 OpenDRIVE

Todo: 目前没有相应的转换工具

13.1 Apollo 如何和 Carla 联合仿真？

想要使用 Carla 和 Apollo 进行自动驾驶的仿真，如何操作？

- 维护者: daohu527@gmail.com
- 版本: 1.0.0
- 日期: 05/19/2024
- 描述:

13.1.1 回答

CHAPTER 14

Dreamview

CHAPTER 15

工具

16.1 如何查看 Apollo 日志？

如何查看 Apollo 日志文件？以及如何打印日志到终端？

- 维护者：daohu527@gmail.com
- 版本：1.0.0
- 日期：05/09/2024
- 描述：

16.1.1 回答

Apollo 提供了五种日志等级：DEBUG、INFO、WARN、ERROR 和 FATAL。默认情况下，系统仅保存 INFO 及以上级别的日志，并仅将 WARN 及以上级别的日志输出到终端。

日志文件存储

Apollo 的日志文件是分模块保存的，它们位于 `data/log` 文件夹中。每次系统启动时，都会根据当前时间戳创建一个新的日志文件，并且会链接到相应的 `ModuleName.INFO.log` 文件下，以便快速访问。历史日志文件则按照时间戳进行保存。

以感知模块为例，当启动感知模块时，系统会基于当前时间戳创建一个新的日志文件，并将其链接到 `perception.INFO.log`。

如果想日志输出到单独的文件，可以使用以下方式，日志会保存到 `ModuleName.INFO.log` 文件中。

```
ALOG_MODULE("ModuleName", INFO) << "Hello world!";
```

终端日志打印

默认情况下, DEBUG 级别的日志是不显示的, 您可以通过修改 `cyber/setup.bash` 文件中的 `GLOG_v` 的值为 4 来启用 DEBUG 日志的显示。

```
# 启用 DEBUG 日志
# for DEBUG log
export GLOG_v=4
```

若您希望日志信息同时输出到终端和日志文件中, 您需要设置 `GLOG_alsologtostderr` 的值为 1。默认情况下, 该值为 0 (即不输出到终端)。

```
#export GLOG_alsologtostderr=0

# 允许日志同时输出到终端
export GLOG_alsologtostderr=1
```

确保在终端中执行了 `source cyber/setup.bash` 命令, 以应用这些修改。

16.2 如何查看 coredump 原因?

程序异常奔溃后, 如何查看 coredump 原因?

- 维护者: daohu527@gmail.com
- 版本: 1.0.0
- 日期: 05/19/2024
- 描述:

16.2.1 回答

程序异常奔溃后, 可以通过 coredump 文件来查看具体的错误原因。

coredump 文件路径

Apollo 的 core dump 文件保存在 `data/core` 目录下, 当程序崩溃的时候, 系统会生成一个 core dump 文件, 通常命名为 `core` 或 `core.pid`, 其中 `pid` 是进程 ID。

查看 coredump 原因

`gdb` (GNU Debugger) 是一个强大的工具, 用于分析 core dump 文件。你需要提供生成 core dump 文件的可执行文件和 core dump 文件本身。以下是具体步骤:

1. 启动 `gdb`, 并加载可执行文件和 core dump 文件:

```
gdb <path-to-executable> <path-to-core-dump>
```

例如:

```
gdb ./my_program data/core/core.12345
```

2. 在 `gdb` 中查看崩溃时的堆栈信息:


```
(gdb) bt
```

bt (backtrace) 命令会显示崩溃时的调用堆栈，有助于定位问题。

示例

假设生成的 core dump 文件是 core.12345，你可以按如下步骤查看：

```
$ gdb mainboard data/core/core.12345
```

在 gdb 提示符下：

```
(gdb) bt
# 会显示调用堆栈
(gdb) info locals
# 显示当前帧的局部变量
(gdb) print some_variable
# 打印变量 some_variable 的值
(gdb) list
# 显示当前执行的代码行
(gdb) list my_program.c:42
# 查看 my_program.c 文件第 42 行的代码
```

通过上述步骤，你可以深入分析程序崩溃的原因，并进行相应的调试和修复。

16.3 如何解决编译依赖下载失败的问题？

Apollo 编译过程中会下载依赖包，由于网络问题导致下载失败，如何解决这个问题？

- 维护者：daohu527@gmail.com
- 版本：1.0.0
- 日期：06/12/2024
- 描述：

16.3.1 回答

bazel 编译缓存

在使用 Bazel 编译项目时，往往需要依赖一些外部库，并通过网络拉取远程代码进行编译。然而，当网络不可用或代码无法拉取时，会导致编译失败。

为了解决离线编译的问题，Bazel 支持从本地路径获取依赖文件。因此，只需将当前工程所依赖的代码包放入本地缓存路径下，就可以避免由于网络问题导致的编译失败。

以下是具体步骤：

1. 下载 Bazel 编译 Apollo 所需的依赖文件：
 - 从 [百度网盘 \(n6j2\)](#) 下载依赖文件。
2. 将下载的文件拷贝到 Bazel 编译缓存目录：

```
# dependency_files 目录包含 Apollo 编译所需的依赖文件
# apollo/.cache/distdir/ 是 Bazel 的缓存路径

cp -r dependency_files/* apollo/.cache/distdir/
```

3. 重新开始编译:

```
./apollo.sh build
```

通过以上步骤, 即可以顺利完成编译。

16.4 拉取 Apollo Docker 失败怎么办?

国内拉取 Apollo Docker 失败如何解决?

- 维护者: daohu527@gmail.com
- 版本: 1.0.0
- 日期: 07/17/2024
- 描述:

16.4.1 回答

由于国内访问 Docker Hub 受限, 作为替代方案, 国内采用百度云镜像下载, 国外继续采用 Docker Hub, 如果提示如下错误信息

```
[INFO] Use default GeoLocation settings
[INFO] Start pulling docker image apolloauto/apollo:dev-x86_64-18.04-20240620_1444 ...
Error response from daemon: manifest for apolloauto/apollo:dev-x86_64-18.04-20240620_
→1444 not found: manifest unknown: manifest unknown
[ERROR] Failed to pull docker image : apolloauto/apollo:dev-x86_64-18.04-20240620_1444
```

错误原因

可以看到, 上述错误信息使用的还是默认的配置 Use default GeoLocation settings, 即 Docker Hub 的方式。

解决方法

因为启动脚本会通过读取系统市区, 如果是北京时间则是国内, 采用国内百度云镜像下载, 否则是国外, 默认采用 Docker Hub, 因此当出现上述错误的时候, 检查当前系统时区是否是北京时区“+0800”。

如何新增车辆配置?

<!--问题描述 (介绍问题的背景)-->

- 维护者: <daohu527@gmail.com>
- 版本: 1.0.0
- 日期: 07/25/2024
- 描述:

回答

<!--回答逻辑清晰，可以采用多级标题，添加图示等方式。-->