



日期： 2023 年 1 月 8 日

成绩： _____

学院： 智能工程学院

课程： 多智能体集群控制

主题： 基于“车—车”协同的避撞控制

专业： 智能科学与技术

姓名： 方桂安

学号： 20354027

1 实验目的及要求

1. 基于前面三个实验，设计一个基于“车—车”协同的避撞控制算法。
2. 搭建一个“闭环控制”的 Simulink 模型验证这个算法。
3. 查阅资料，拓展实验内容，学习更多关于多智能体感知的相关知识。

2 实验方法与步骤

2.1 例程

首先我打算先跑通老师提供的例程，参考、消化其中的内容。

1. 运行 ScenarioAnalysis.slx，运行仿真模拟的结果是生成一个 egoCar1.mat。

在 ScenarioAnalysis 中，主要有两个模块 Scenario Reader 和 VehicleToWorld，再加上其他 function 的处理之后，仿真会读取 My_Scenario.mat 中的场景，最后导出 egoCar1.mat，这个文件中包含了仿真的场景信息，包括车辆的位置、速度、ID 等信息。

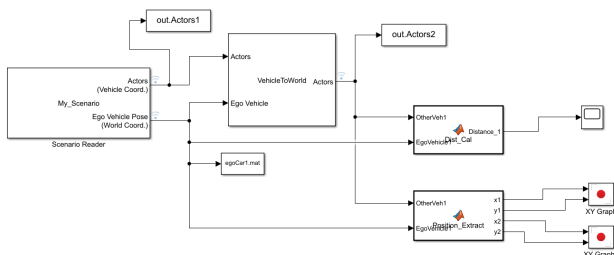


图 1: ScenarioAnalysis.slx 仿真模型

结合前几节课的知识，我先使用 drivingScenarioDesigner ('My_Scenario.mat') 查看原始场景及其仿真效果。

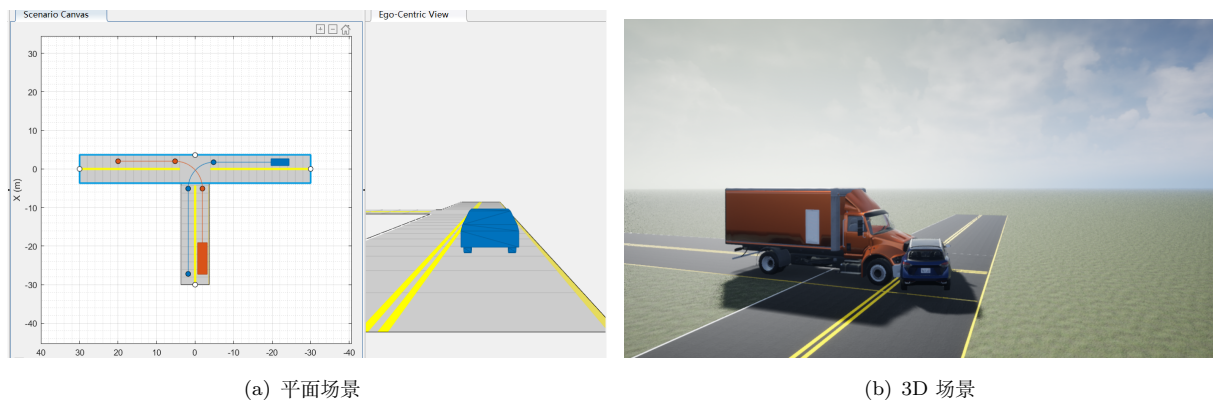


图 2: 仿真结果

显然两辆车不加入避障控制是会相撞的。另外结合老师的提醒，仿真运行结束后需要打开总线编辑器，修改变量名。

2. 运行 CollisionAvoidance_Demo.slx 中的“Run Setup Script”, 其实就是运行 Extract_Actors_Param.m 文件，会从内存区域读入 EgoCar 的路径规划，将里面的数据分解为各个姿态信息。
3. 运行 CollisionAvoidance_Demo.slx 本体的仿真模型，Dist_Cal 函数会按照设定的安全距离阈值闭环控制两小车的速度，从而避免相撞。
4. 运行 CollisionAvoidance_Demo.slx 中的“Visualization of Outputs”, 其实就是运行 Plot_Vehicles.m 文件，会将仿真结果绘制出来，其中密集的点代表对应的车辆减速了。

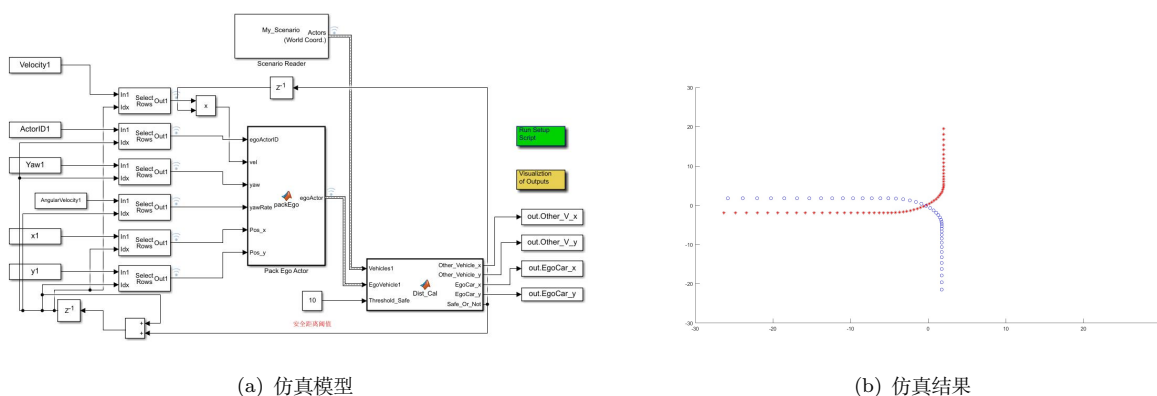


图 3: CollisionAvoidance_Demo.slx

2.2 拓展

得幸于深度学习猛烈的发展，自动驾驶感知系统在近几年有了飞速的提升。尽管如此，单车的感知系统仍有其无法克服的困难。当自动驾驶汽车在遇到严重的遮挡或距离较远的物体时，往往不能有很好的表现。这种情况往往是由单车传感器在遮挡处或是远处信息太稀疏导致的，很难被算法克服。针对这种单车物理上的限制，工业界和学术界开始研究如何利用多车之间的视觉信息传递来彻底解决遮挡问题并增加感知视野。如图4所示，ego 汽车（绿色框）正在一个 T 型的交叉口准备左转，此时正有两辆车向它这个方向快速驶来，但是由于停在路边的汽车，它的激光雷达根本无法感知到这两辆驶来的汽车（如图4中间所示），很可能会造成重大事故。但幸好此时有另外一辆自动驾驶汽车在另一侧经过（图4左被蓝色方框标出的 CAV1），它能很好地捕捉到这两辆驶来汽车的信息（图4右所示），并且将该信息传递给 ego，从而避免车祸的发生。



图 4: 协同感知效果展示

本次实验中老师希望我们加入 V2X 通信模块，使得 EgoVehicle 能通过通信模块来获取附近的车辆的位置信息或者加入车辆的运动学模块或者感知模块来实现“车—车”协同的避撞控制。通过查阅资料，我决定使用 OpenCOOD 来实现上述目标。

2.3 OpenCOOD

OpenCOOD 是首个用于自动驾驶的协同感知代码框架，该代码框架的主要特征有：

- 提供了一套简易于使用的 API，方便用户读取 OPV2V 数据并转化为相应的格式供 pytorch 模型直接使用；
- 提供了多个 SOTA 3D 点云检测模型代码，包括 PointPillar, VoxelNet, Pixor, SECOND；
- 支持多种常见地多智能体融合策略，包括后融合，前融合，中间层融合；
- 提供多种协同感知 SOTA 模型，包括 Cooper, F-Cooper, Attentive Fusion 等，并且作者承诺会持续更新所有市面上的最新算法；
- 提供使用的 log replay 工具来回放 OPV2V 数据，并支持用户在不改变原数据的基础上增加新事件；
- 代码库还提供了详细的文档和 tutorial, 方便用户更好理解。

2.4 V2X-ViT

2.4.1 预备知识

1. V2X:

V2X 是指车载通信 (vehicle-to-everything, V2X) 技术。它是一种利用无线通信技术使汽车、行人、自行车、路面设施、基础设施和其他交通工具之间能够相互通信的技术。

V2X 技术可以分为两类：基于无线局域网 (WLAN) 的 V2X 技术 (如 IEEE 802.11p) 和基于移动网络的 V2X 技术 (如 LTE-V)。前者可以使汽车之间相互通信，后者则可以使汽车与基础设施之间相互通信。

V2X 技术可以应用于许多不同的场景，包括交通信号控制、路况预测、自动驾驶等。它有助于提高交通流畅度，降低交通事故的发生率，并为司机提供重要的信息。

2. ViT:

Vision Transformer (ViT) 是一种用于计算机视觉任务的深度学习模型。它主要由两部分组成：Transformer 模型和卷积层。

Transformer 模型是一种基于注意力机制的深度学习模型，在自然语言处理任务中非常流行。它通过自注意力机制来处理序列数据，不需要依赖长度固定的序列。

ViT 通过将图像划分成若干个像素块，并将每个像素块视为一个单独的“令牌” (即一个序列中的元素)，然后使用 Transformer 模型来处理这些“令牌”。在此过程中，Transformer 模型会学习如何自动对像素块进行分组，从而自动提取图像中的有用特征。

ViT 还包含一个卷积层，它可以在 Transformer 模型输出之后将结果进一步处理，从而提高模型的准确度。

ViT 在计算机视觉任务中表现出色，尤其是在图像分类和对象检测等任务中。它的出现也推动了自然语言处理和计算机视觉领域的交叉研究。

2.4.2 使用 Vision Transformer 的 Vehicle-to-Everything 协作感知

最终我选择了这篇 ECCV 2022 的论文复现来实现要求的车车协同避撞控制。

算法流程解析：

1. **元数据共享**: 在协作的早期阶段，每个 agent 在通信网络中共享元数据，例如姿势、外在和 agent 类型 (表示基础设施或车辆)。选择其中一个连接的 AV 作为自车辆 (e) 以在其周围构建 V2X 图，其中节点是 AV 或基础设施，边缘表示定向 V2X 通信通道。在这项工作中关注特征共享延迟并忽略元数据共享延迟，因为它的尺寸最小。更具体地说，假设元数据的传输是同步良好的，这意味着每个 agent i 都可以在时间接收到自姿势。在接收到本车的姿态后，附近的所有其他连接 agent 将在特征提取之前将自己的 LiDAR 点云投影到本车的坐标系中。

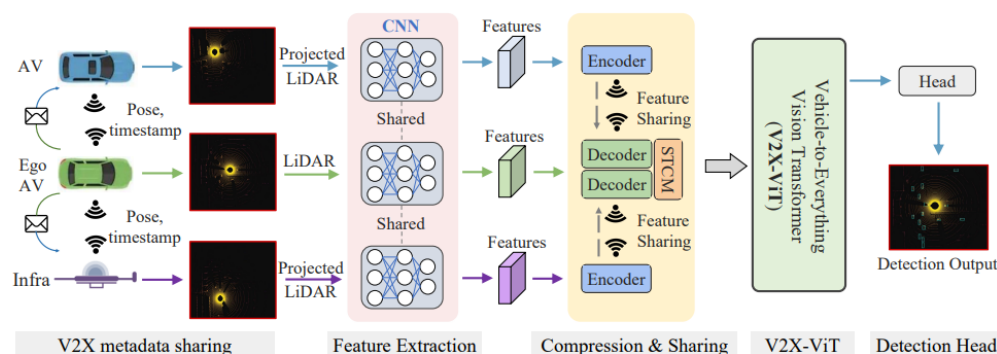


图 5: V2X-ViT 流程图

2. **特征提取:** 利用 PointPillar 从点云中提取视觉特征 (PointPillar 具有低推理延迟和优化的内存使用)。原始点云将被转换为一个堆叠的 pillar 张量, 然后分散到一个二维伪图像, 并输入到 PointPillar backbone。backbone 提取信息特征图, 表示 agent i 在时间的特征, 高度为 H 、宽度为 W 和通道 C 。

3. **压缩与共享:** 为了减少所需的传输带宽, 利用一系列 1×1 卷积沿通道维度逐步压缩特征图。然后将大小为 (H, W, C') 的压缩特征 (其中 $C' \ll C$) 传输到自车辆 (e), 在其上使用 1×1 卷积将特征投影回 (H, W, C) 。

在激光雷达数据被连接的 agent 捕获时和被提取的特征被自载体接收时之间存在着不可避免的时间间隔。因此, 从周围 agent 收集的特征往往与自载体上捕获的特征在时间不一致。为了纠正这种延迟引起的全局空间失调, 需要将接收到的特征转换 (即旋转和平移) 为当前的自车辆的姿态。因此, 利用了一个时空校正模块 (STCM), 它使用了一个差分变换和采样算子 Γ 来对特征映射进行空间扭曲。还计算了一个 ROI mask, 以防止网络关注由空间扭曲引起的填充零。

4. **V2X-ViT:** 从连接 agents 聚合的中间特征, 被输入框架的主要组件, 即 V2X-ViT, 以使用自注意力机制进行迭代 agents 间和 agents 内特征融合。在整个 Transformer 中将特征图保持在相同的高分辨率水平, 因为作者观察到缺乏高清特征会极大地损害模板检测性能。

5. **检测层:** 在收到最终的融合特征图后, 应用 2 个 1×1 卷积层进行框回归和分类。回归输出为 $(x, y, z, w, l, h, \theta)$, 分别表示预定义 Anchor box 的位置、大小和偏航角。分类输出是作为每个 Anchor box 的对象或背景的置信度得分。这里使用 PointPillar 中定义的相同损失, 即用于回归的 Smooth L1 Loss 和用于分类的 Focal Loss。

作者的目标是设计一个定制的 Vision Transformer, 可以共同应对常见的 V2X 挑战。首先, 为了有效地捕捉基础设施和 AV 之间的异构图表示, 作者构建了一个异构多智能体自注意力模块, 该模块根据节点和边类型学习不同的关系提出了一种新颖的空间注意力模块, 即多尺度窗口注意力 (MSwin), 它可以捕获各种尺度的远程交互。MSwin 使用多个窗口大小来聚合空间信息, 这大大提高了对定位错误

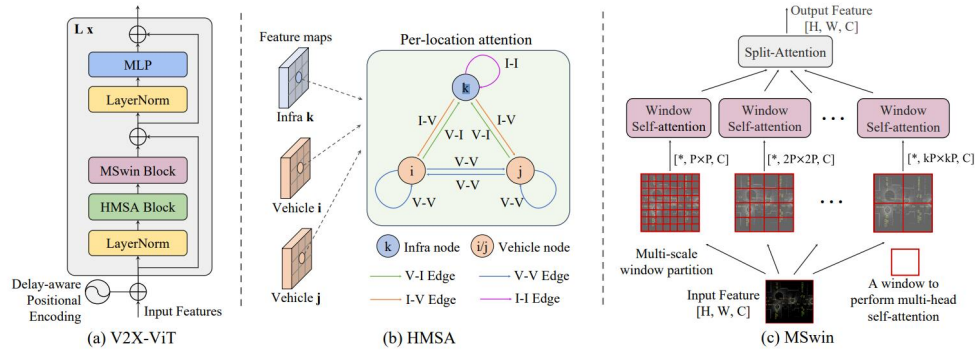


图 6: V2X-ViT 结构图

的检测鲁棒性。最后，这 2 个注意力模块以分解的方式集成到单个 V2X-ViT 块中（如图6a 所示）能够在整个过程中保持高分辨率特征。堆叠了一系列 V2X-ViT 块以迭代地学习 agent 间交互和每个 agent 的空间注意力，从而产生用于检测的强大聚合特征表示。

3 实验结果与心得

3.1 实验结果

首先使用 readme 中的指示安装好必备的环境，方法如下：

```

1  # Clone repo
2  git clone https://github.com/DerrickXuNu/v2x-vit
3  cd v2x-vit
4
5  # Setup conda environment
6  conda create -y --name v2xvit python=3.7
7  conda activate v2xvit
8  # pytorch ≥1.8.1, newest version can work well
9  conda install -y pytorch torchvision cudatoolkit=11.3 -c pytorch
10 # spconv 2.0 install, choose the correct cuda version for you
11 pip install spconv-cu113
12
13 # Install dependencies
14 pip install -r requirements.txt
15 # Install bbx nms calculation cuda version
16 python v2xvit/utils/setup.py build_ext --inplace
17
18 # install v2xvit into the environment
19 python setup.py develop

```


另外还需要安装 CARLA，CARLA 提供了一个可以模拟真实世界的环境，并且可以用来构建、测试和评估自动驾驶汽车的控制系统。注意版本必须是 0.9.12，方法如下：

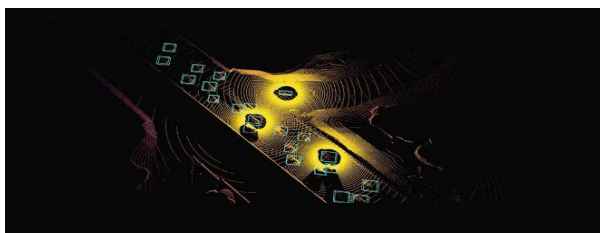
```
1 sudo apt-get update # Update the Debian package index
2 sudo apt-get install carla-simulator=0.9.12-1
3 # In this case, "0.9.12" refers to a CARLA version, and "1" to the Debian revision
4 cd /opt/carla-simulator # Open the folder where CARLA is installed
```

由于我的笔记本显卡算力和显存较差，无法在 67G 数据集上训练自己的模型，故我在 google drive 上下载了 20G 的测试集和训练了 60 个 epoch 的预训练模型，然后把它放在 v2x-vit/logs/v2x-vit 下。将 v2x-vit/logs/v2x-vit/config.yaml 中的 validate_path 改为 'v2xset/test'。

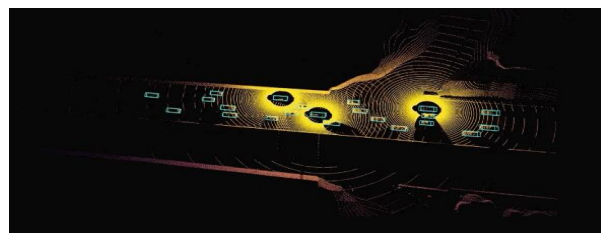
最后,运行下面的命令来进行测试:python v2xvit/tools/inference.py -model_dir CHECKPOINT_FOLDER -fusion_method FUSION_STRATEGY [-show_vis] [-show_sequence]。

参数解释：

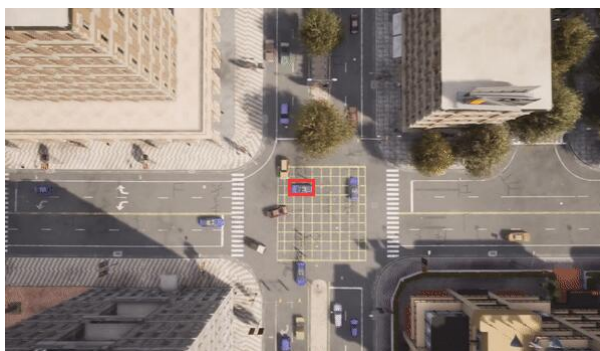
- model_dir：你保存的模型的路径。
- fusion_method：表示融合策略，目前支持“early”、“late ”和“intermediate”。
- show_vis：是否将检测结果与点云进行可视化叠加。
- show_sequence：检测结果将在视频流中可视化。它不能与 show_vis 同时设置。



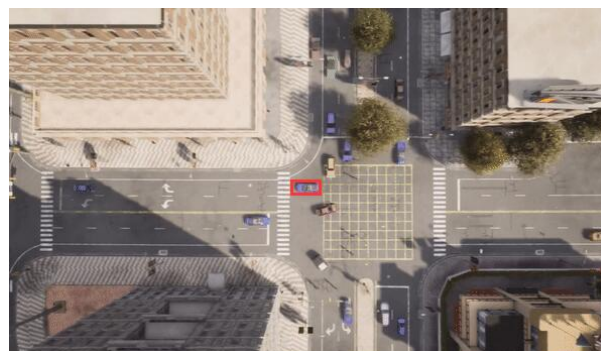
(a) 拐弯前



(b) 拐弯后



(c) 通过路口前



(d) 通过路口后

图 7: 仿真结果

不难看出，在 V2V 通信感知的帮助下，算法在多个场景都表现良好，小车反馈的信息加上车-车协同的信息进入模型融合后反馈，从而使得控制算法更加准确，形成了闭环来避免碰撞。

3.2 实验心得

准确感知复杂的驾驶环境对于自动驾驶汽车 (AV) 的安全至关重要。随着深度学习的最新进展，单车感知系统的鲁棒性在语义分割和模板检测等多项任务中表现出显著改善。尽管最近取得了进展，但挑战依然存在。单智能体感知系统往往会遭受远距离的遮挡和稀疏的传感器观察，这可能会导致灾难性的后果。造成这种问题的原因是单个车辆只能从单一视角感知环境，视野有限。为了解决这些问题，最近的研究通过调查 Vehicle-to-Vehicle (V2V) 协作来利用同一场景的多个视点的优势，其中视觉信息（例如，检测输出、原始感官信息、中间深度学习特征）来自多个附近的 AV 共享，以便全面准确地了解环境。

尽管 V2V 技术有望彻底改变移动行业，但它忽略了一个关键——路边基础设施。自动驾驶汽车的存在通常是不可预测的，而一旦安装在十字路口和人行横道等关键场景中，基础设施总是可以提供支持。此外，在较高位置配备传感器的基础设施具有更广阔的视野和可能更少的遮挡。

尽管有这些优势，包括部署强大的 V2X 感知系统的基础设施并非易事。与所有代理都是同质的 V2V 协作不同，V2X 系统通常涉及由基础设施和 AV 形成的异构图。基础设施和车辆传感器之间的配置差异，例如类型、噪音水平、安装高度，甚至传感器属性和模式，使 V2X 感知系统的设计具有挑战性。此外，GPS 定位噪声和自动驾驶汽车和基础设施的异步传感器测量可能会引入不准确的坐标转换和滞后的传感信息。未能妥善处理这些挑战将使自动驾驶系统变得十分脆弱。

本次实验我在跑通老师提供的示范例程的基础上，利用 OpenCOOD 框架复现了 sota 项目 V2X-ViT，通过一个统一的融合框架，用于 V2X 感知，可以共同应对上述挑战。该模型在 paperwithcode 的 V2XSet 榜单上排名第一，AP0.5 分数达到了 0.882。

通过测试的可视化，我也成功实现了基于“车—车”协同的避撞控制。通过本次实验，我对 V2V，乃至 V2X 的感知系统有了更深的理解，在实践中更深刻地体会到了多智能体集群控制的深刻意义。

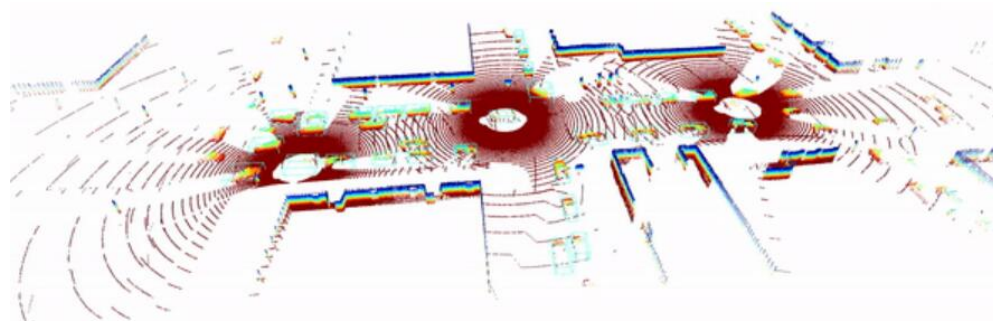


图 8: 多智能体集群控制