深蓝学院
shenlanxueyuan.com

第4章 多传感器融合算法

本章内容

1. 融合问题定义及背景
2. 后融合算法设计与实现
3. 前融合算法设计与实现
4. 场景理解简介
5. 融合系统实现

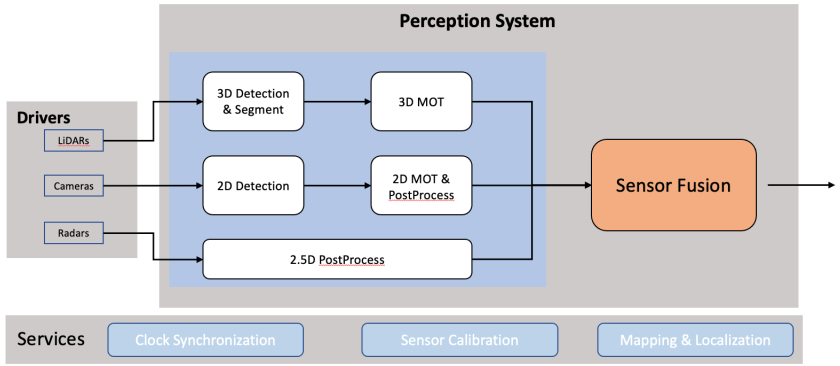
本章内容

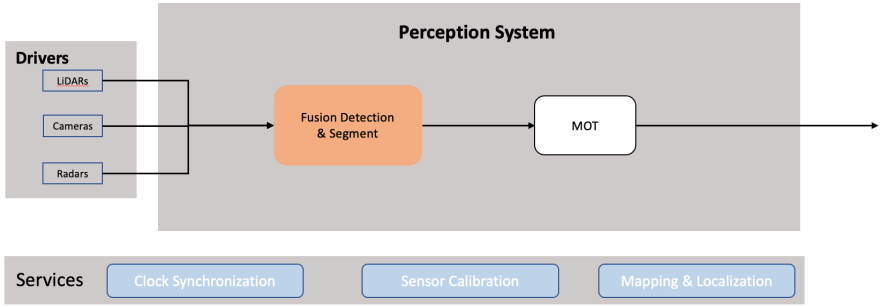
1. 融合问题定义及背景
2. 后融合算法设计与实现
3. 前融合算法设计与实现
    1. DeepLearning及前融合介绍
    2. Camera/LiDAR/Radar之间的前融合
    3. 前融合与后融合的关系
4. 场景理解简介
5. 融合系统实现

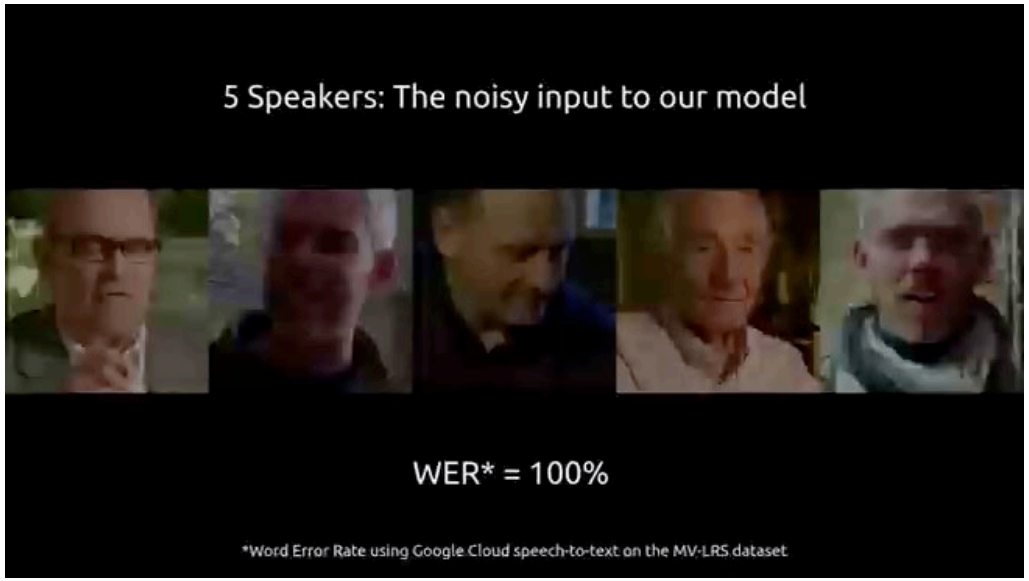
本章内容

1. 融合问题定义及背景
2. 后融合算法设计与实现
3. 前融合算法设计与实现
    1. Deep Learning及前融合介绍
    2. Camera/LiDAR/Radar之间的前融合
    3. 前融合与后融合的关系
4. 场景理解简介
5. 融合系统实现

关于前融合







多模态融合

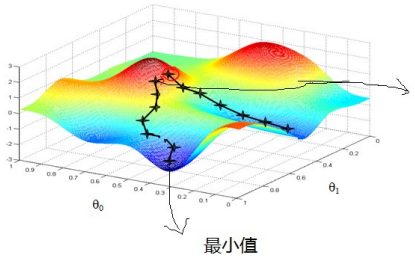
- 模态（modality）是指信息的某种特定的表达形式，比如语音、图像或文本。
- Deep Learning是多模态融合的主要技术手段

Deep Learning介绍

梯度下降

$$\min_{\boldsymbol{x}} \frac{1}{2}\|f(\boldsymbol{x})\|_2^2$$

其中，$\boldsymbol{x} \in R^n$，f是任意非线性函数，$f(\boldsymbol{x}) \in R^m$



求解思路：

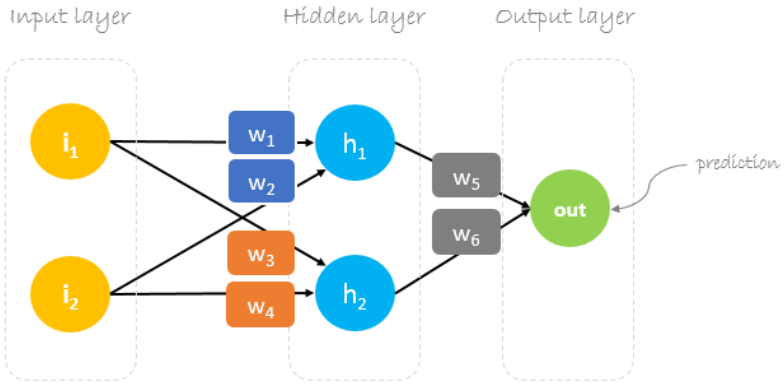
(1) 给定某个初始值$\boldsymbol{x}_0$；

(2) 对于第k次迭代，寻找一个增量$\Delta\boldsymbol{x}_k$，使得$\|f(\boldsymbol{x}_k + \Delta\boldsymbol{x}_k)\|_2^2$达到极小值；

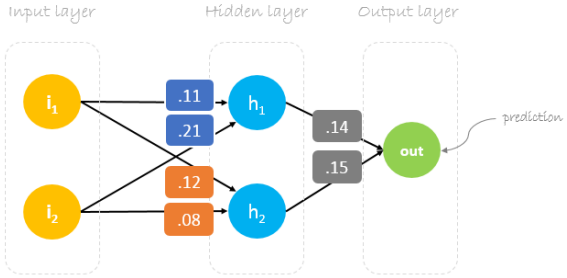
(3) 若$\Delta\boldsymbol{x}_k$足够小，则停止；

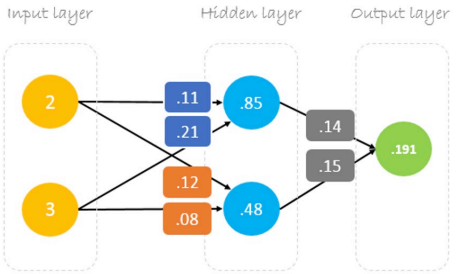
(4) 否则，令$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \Delta\boldsymbol{x}_k$，返回第(2)步。

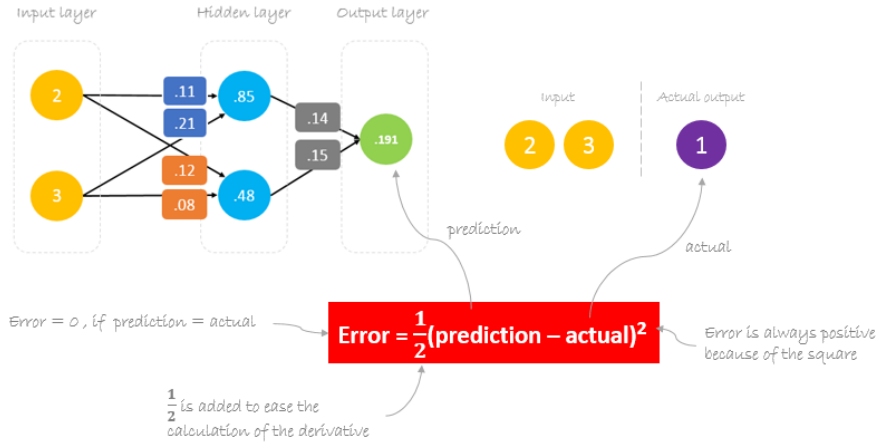
神经网络



$$out = [w_5 \quad w_6]\begin{bmatrix} w_1 & w_2 \\ w_3 & w_4 \end{bmatrix}\begin{bmatrix} i_1 \\ i_2 \end{bmatrix} = w_5(w_1 i_1 + w_2 i_2) + w_6(w_3 i_1 + w_4 i_2)$$







Error = $\frac{1}{2}(0.191 - 1.0)^2 = 0.327$

$$\frac{\partial error}{\partial w_6} = \frac{\partial error}{\partial pred}\frac{\partial pred}{\partial w_6}$$

$$= \frac{\partial error}{\partial pred}(w_3 i_1 + w_4 i_2)$$

$$= (pred - actual)\frac{\partial (pred - actual)}{\partial pred}(w_3 i_1 + w_4 i_2)$$

$$= (pred - actual)h_2 = \Delta h_2$$

$$\frac{\partial error}{\partial w_1} = \frac{\partial error}{\partial pred}\frac{\partial pred}{\partial h_1}\frac{\partial h_1}{\partial w_1}$$

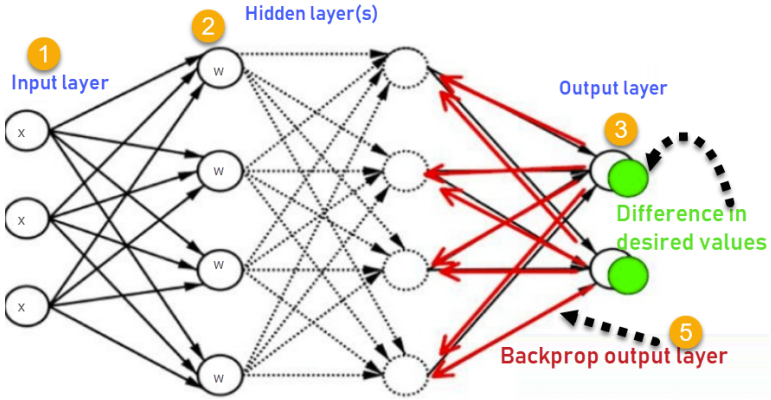
$$= (pred - actual)w_5 i_1$$

$$= \Delta w_5 i_1$$

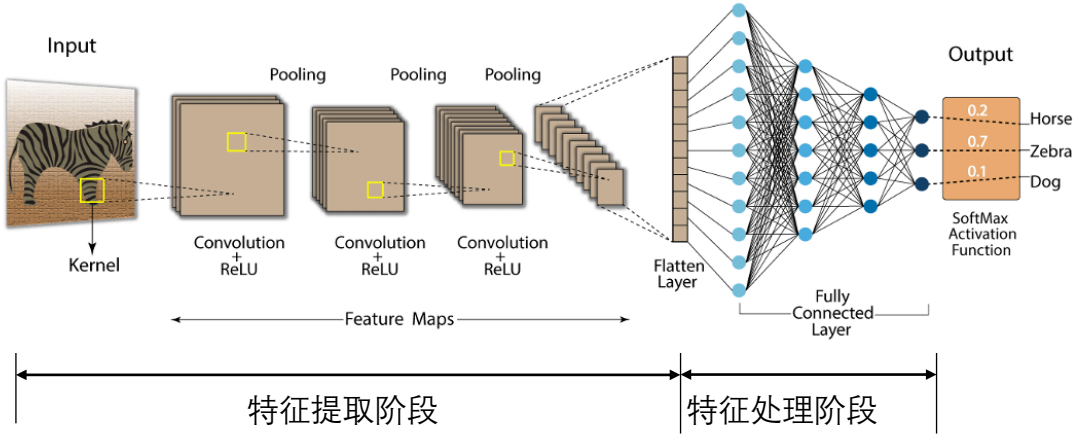
$w_6^* = w_6 - \lambda\Delta h_2$ ... $w_1^* = w_1 - \lambda\Delta w_5 i_1$

Deep Learning介绍

- 神经网络——ForwardPropagation & BackPropagation



- Convolutional Nerual Network



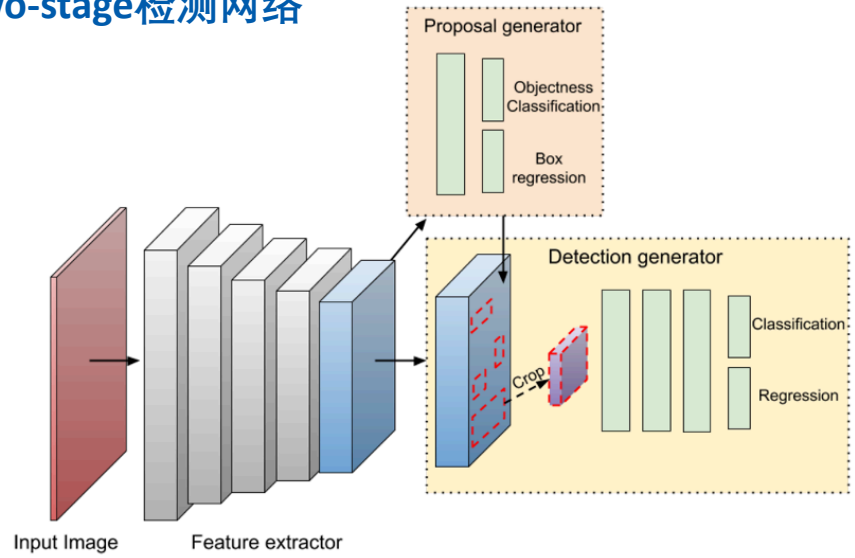
- Convolutions

	0	1	0
1	-4	1	
	0	1	0

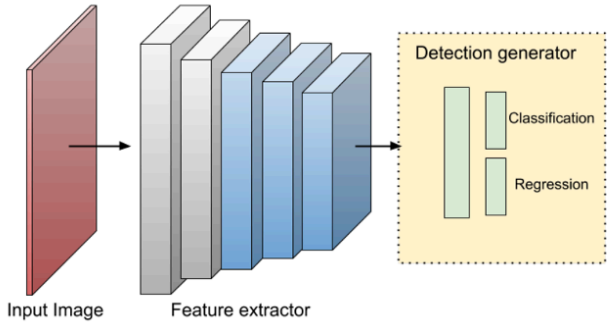
convolutions

CNN在图像模态下的使用

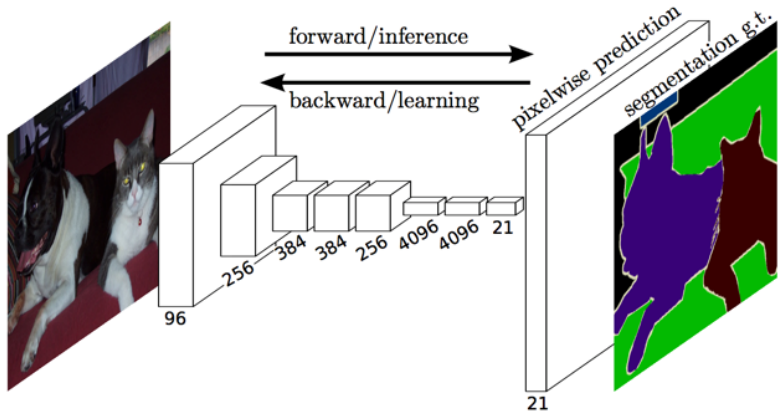
- two-stage检测网络



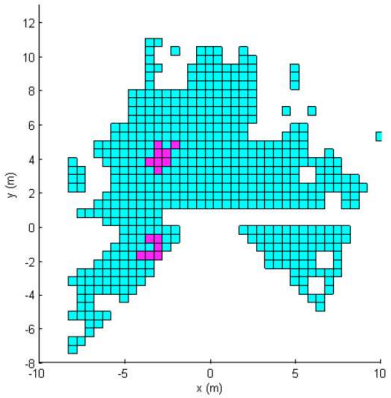
- one-stage检测网络

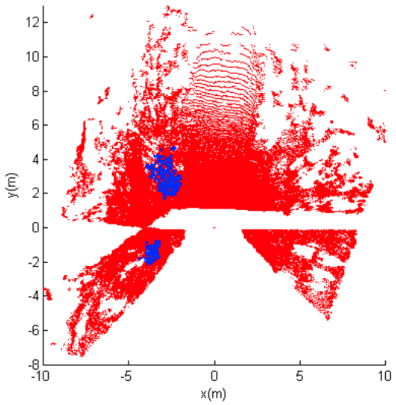


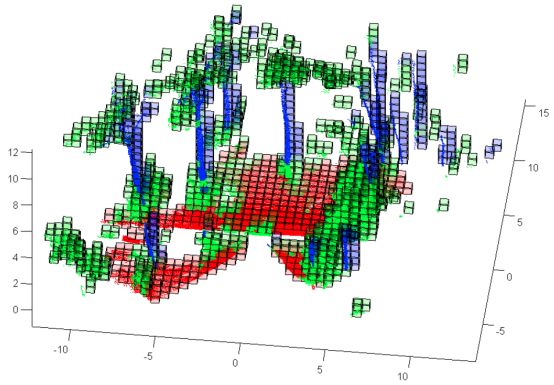
- 分割网络

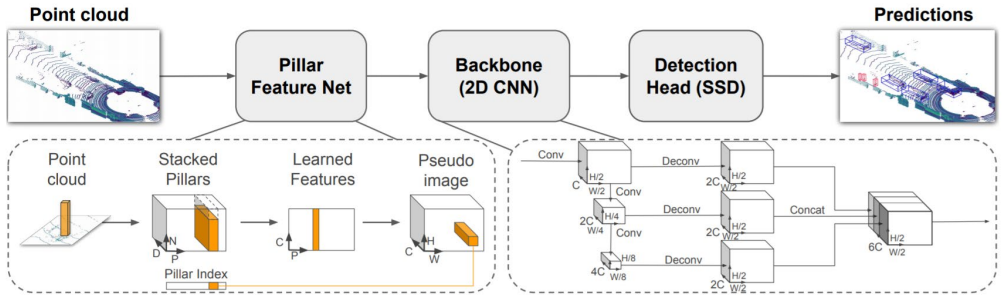


CNN在点云模态下的使用

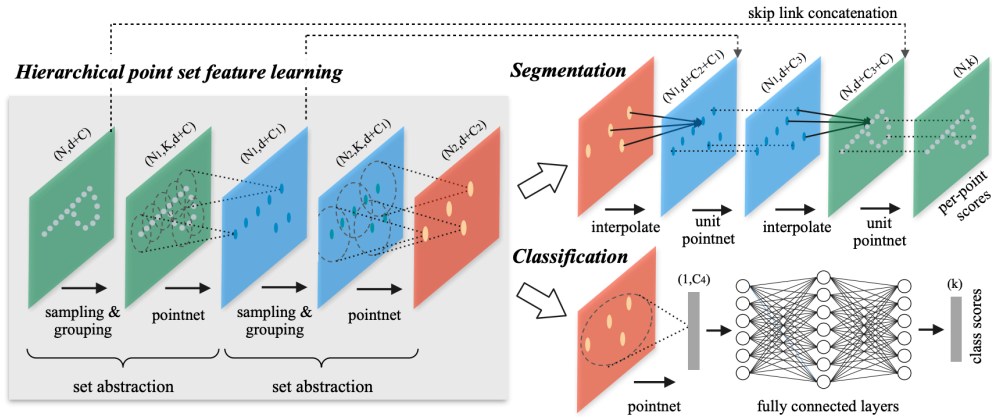




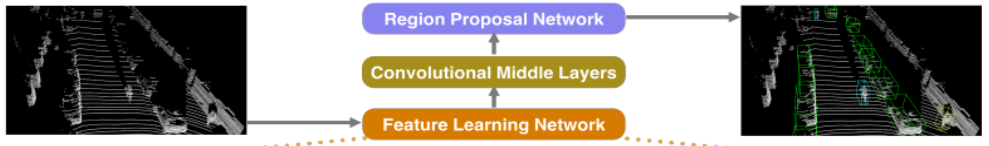


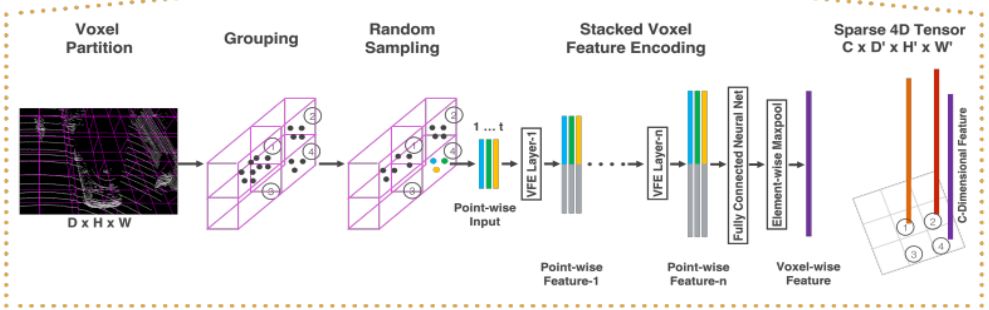


- 模型速度快
- 丢失一些3D信息



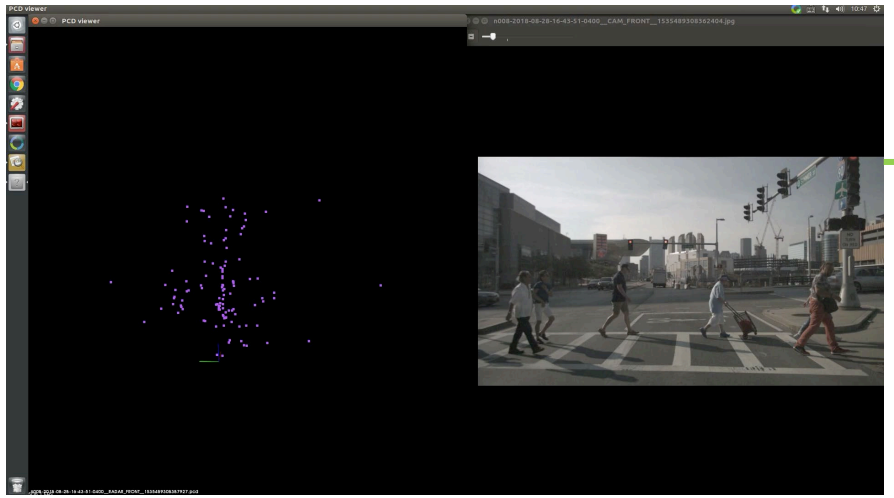
- 细节特征提取好
- 计算量较大





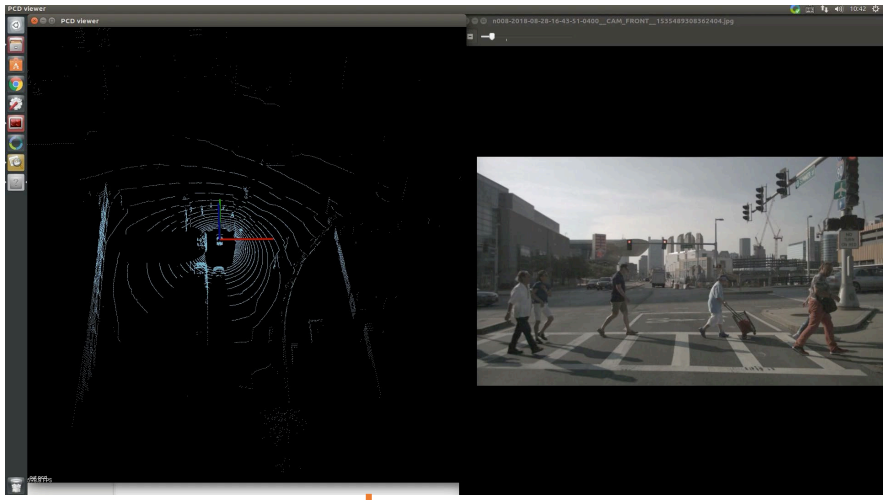
- 精度高
- 计算量大

自动驾驶中的多模态融合：LiDAR/Camera/Radar



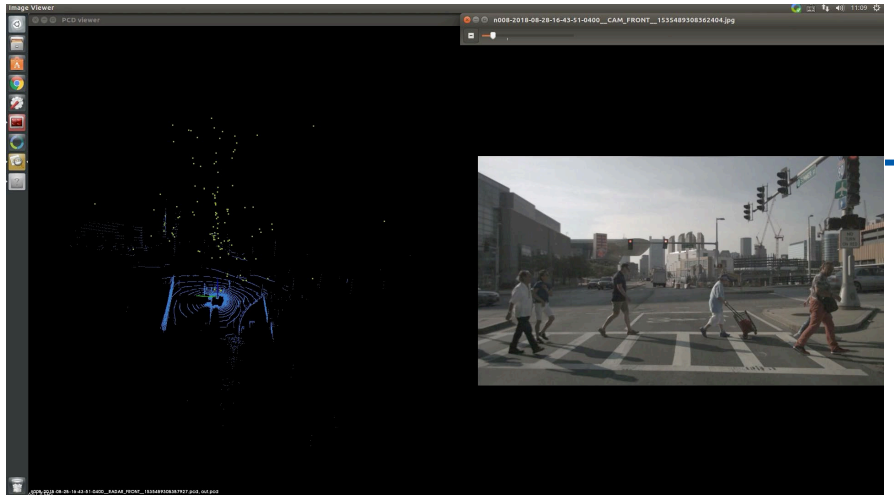
Camera + Radar

- Radar without height info
- Long range with robustness
- Low recall for nearby non-metal obj
- Hight recall for vehicles but over-segment
- Sparse radar points



Camera + LiDAR

- LiDAR with all 3D infos
- Mid-range(~100m)
- High precision but low robustness
- Sparse lidar points



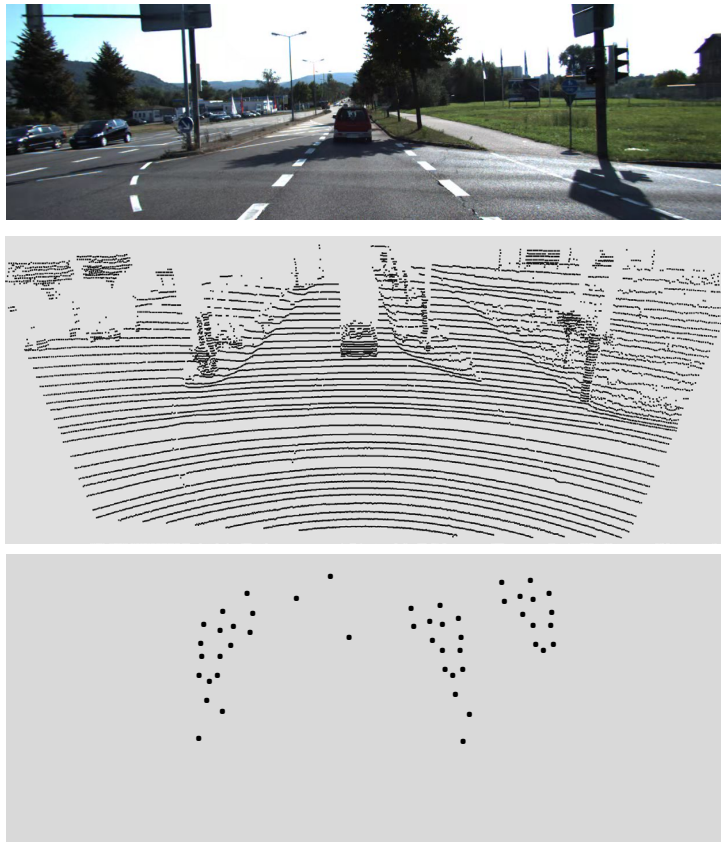
Camera + LiDAR + Radar

- LiDAR with all 3D infos
- Up to ~200m

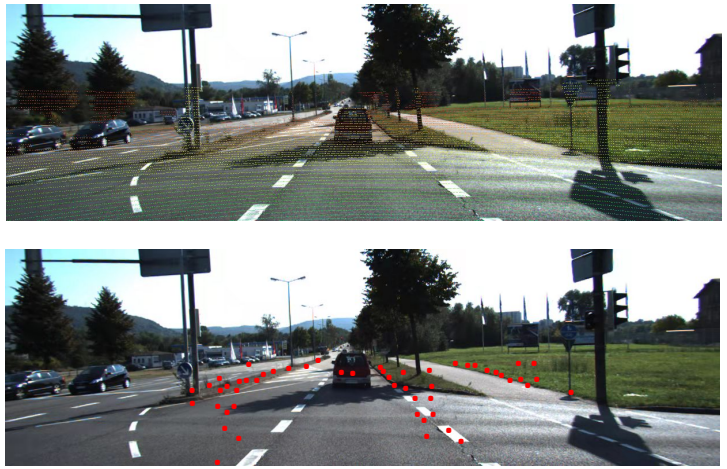
本章内容

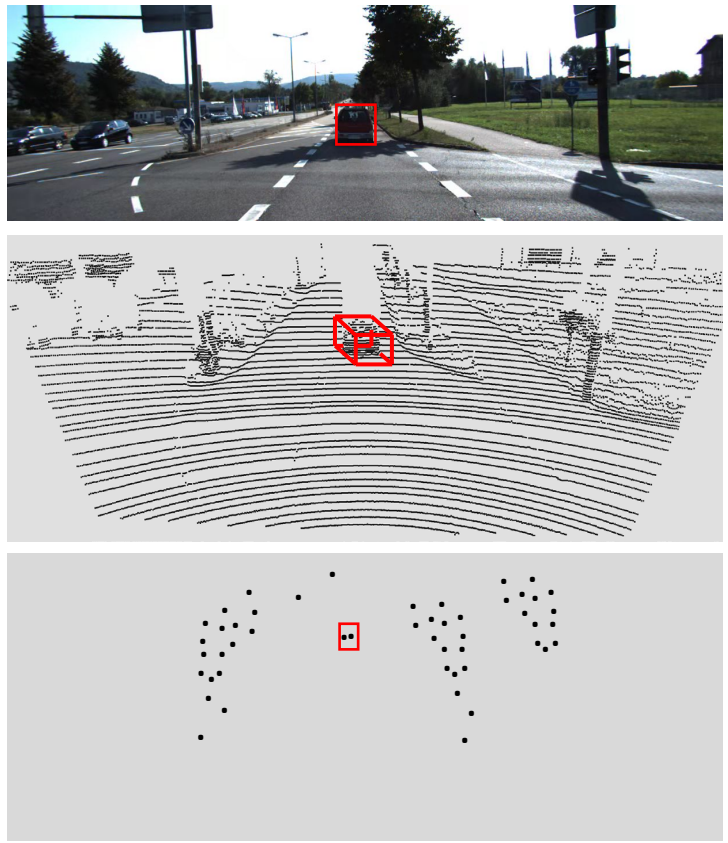
1. 融合问题定义及背景
2. 后融合算法设计与实现
3. 前融合算法设计与实现
    1. DeepLearning及前融合介绍
    2. Camera/LiDAR/Radar之间的前融合
    3. 前融合与后融合的关系
4. 场景理解简介
5. 融合系统实现

如何实现多模态融合

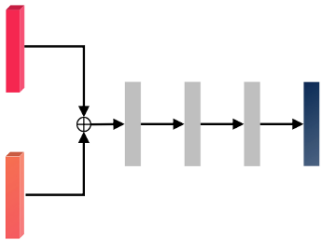


- 单模态特征如何表达?
- 多模态时空如何对齐?
- 多模态特征如何有效融合?

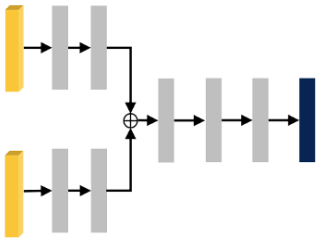




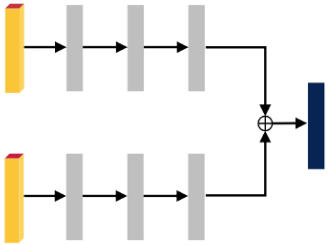
数据融合



特征融合



目标融合

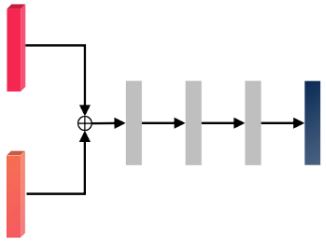


融合模式:

- LiDAR + Camera
- Camera + Radar

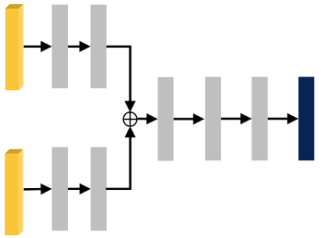
如何实现多模态融合

数据融合



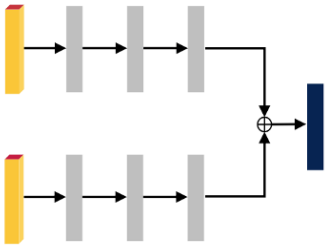
- 要求原始数据时空对齐程度高，像素级和点级别的对齐

特征融合



- 允许有一定程度的不对齐，通常要求时间误差<10ms

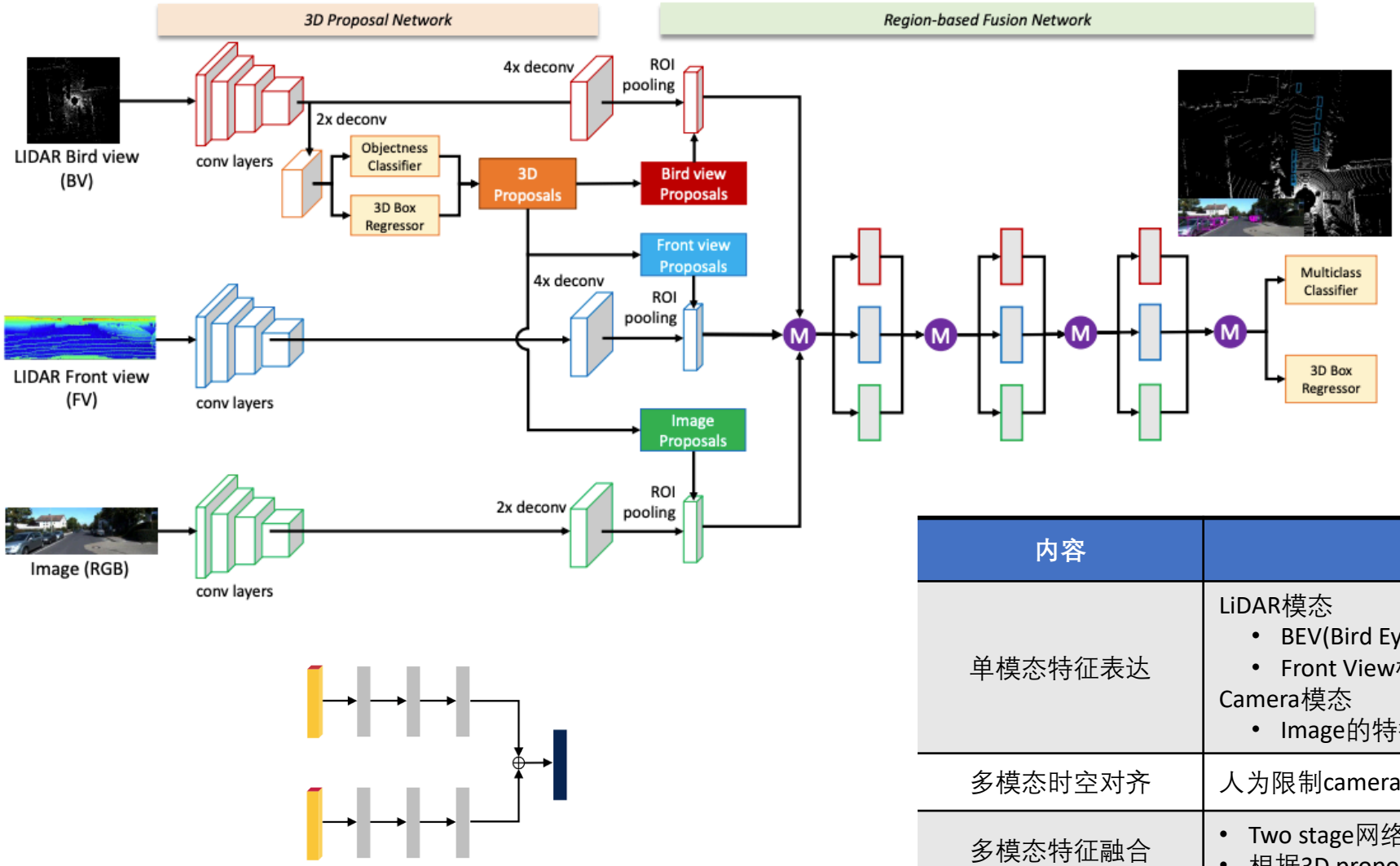
目标融合



- 允许有一定程度的不对齐，通常要求时间误差<10ms

- 单模态特征如何表达？
- 多模态时空如何对齐？
- 多模态特征如何有效融合？

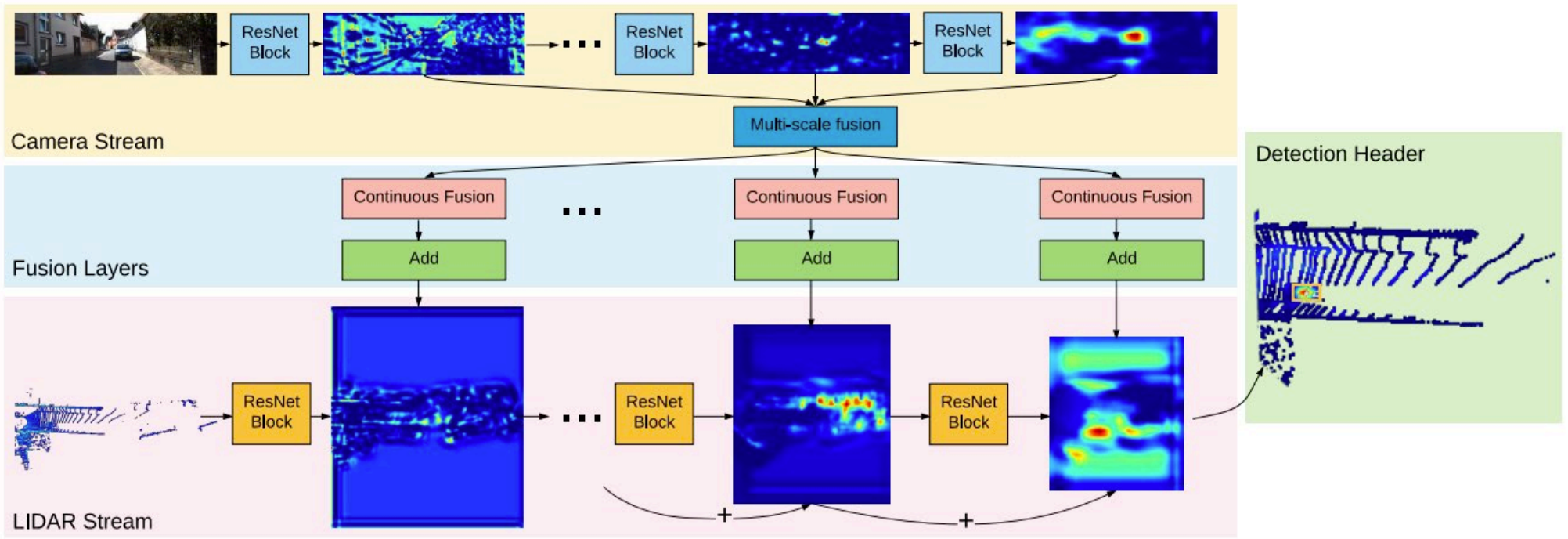
LiDAR+Camera目标融合

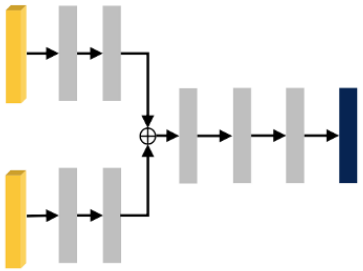


内容	细节描述
单模态特征表达	LiDAR模态 • BEV(Bird Eye View)模式下的特征 • Front View模式下的特征 Camera模态 • Image的特征
多模态时空对齐	人为限制camera和LiDAR的误差在10ms以内
多模态特征融合	• Two stage网络，利用BEV下的特征生成3D proposals • 根据3D proposals汇聚3个维度的特征

[1] MV3D: Multi-View 3D Object Detection Network for Autonomous Driving, CVPR 2017, https://arxiv.org/abs/1611.07759
[2] Code: https://github.com/bostondiditeam/MV3D

LiDAR+Camera特征融合算法



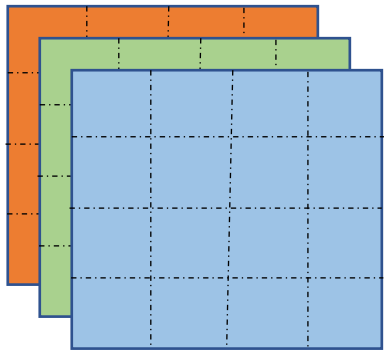


内容	细节描述
单模态特征表达	LiDAR模态 • Front View模式下的特征 Camera模态 • Image的特征
多模态时空对齐	未考虑
多模态特征融合	• 对于LiDAR BEV的每个像素空间，在image中寻找对应的特征向量； • 根据对应的特征向量，预测该像素的特征向量； • 构成的dense特征向量，按照PointPillars的检测方式进行检测。

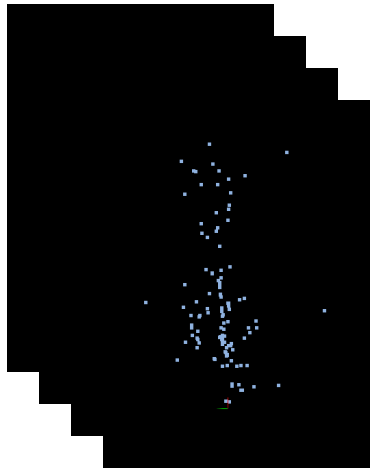
[1]ContFuse: Deep Continuous Fusion for Multi-Sensor 3D Object Detection

Camera+Radar目标融合算法

RGB

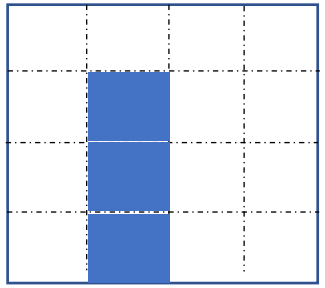


Radar Preprocessing

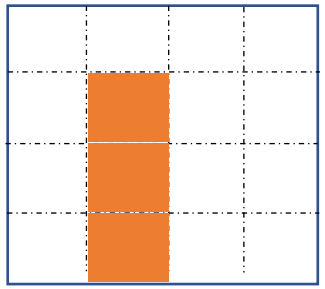


10 cycles

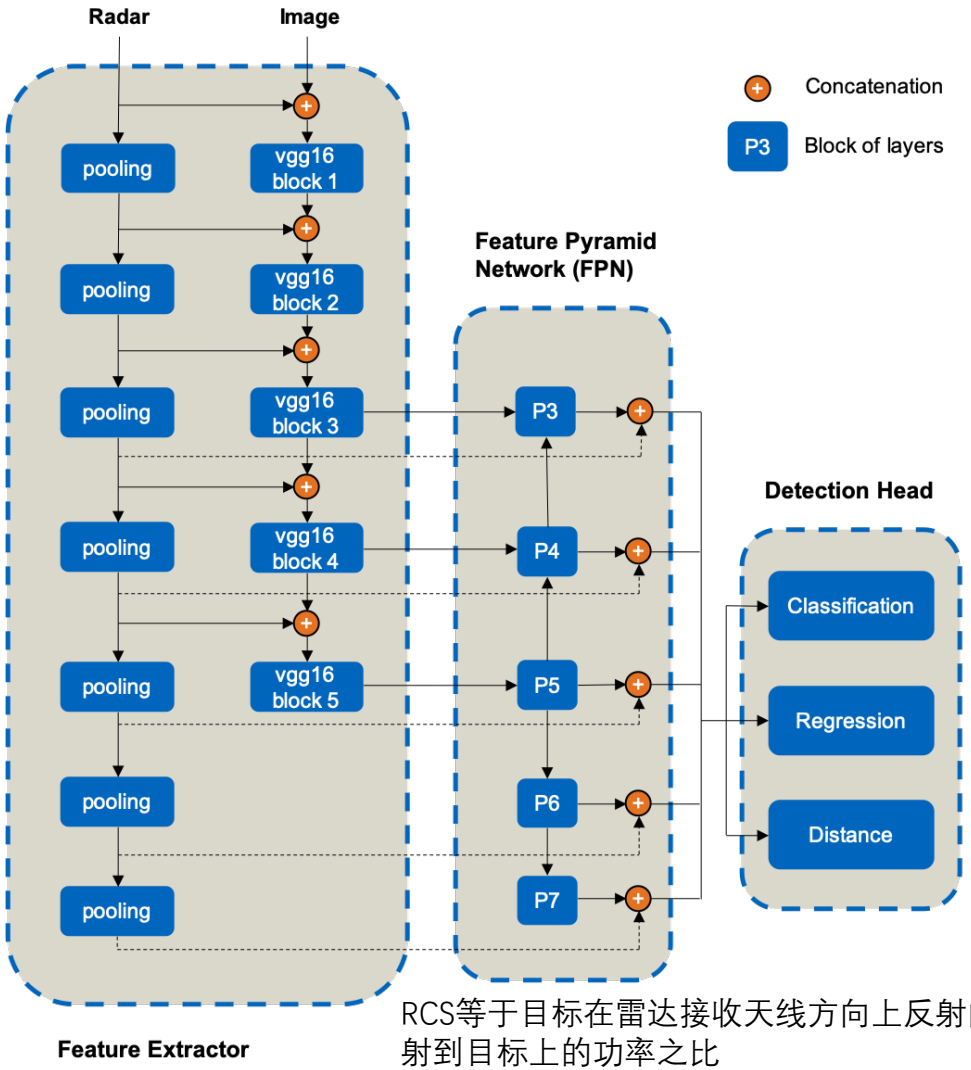
distance



RCS



Network Architecture



RCS等于目标在雷达接收天线方向上反射的功率与入射到目标上的功率之比

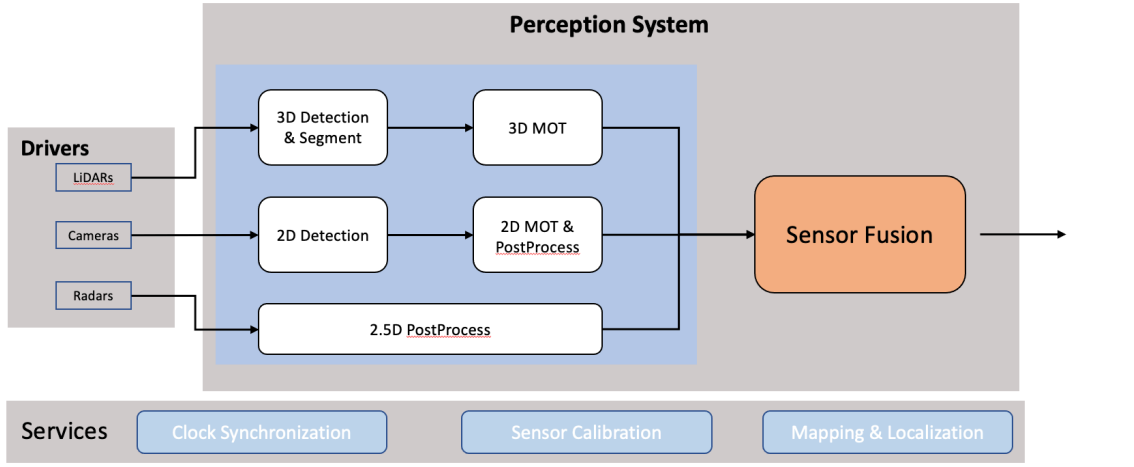
[1] A Deep Learning-based Radar and Camera Sensor Fusion Architecture for Object Detection, https://arxiv.org/abs/2005.07431
[2] Code https://github.com/TUMFTM/CameraRadarFusionNet

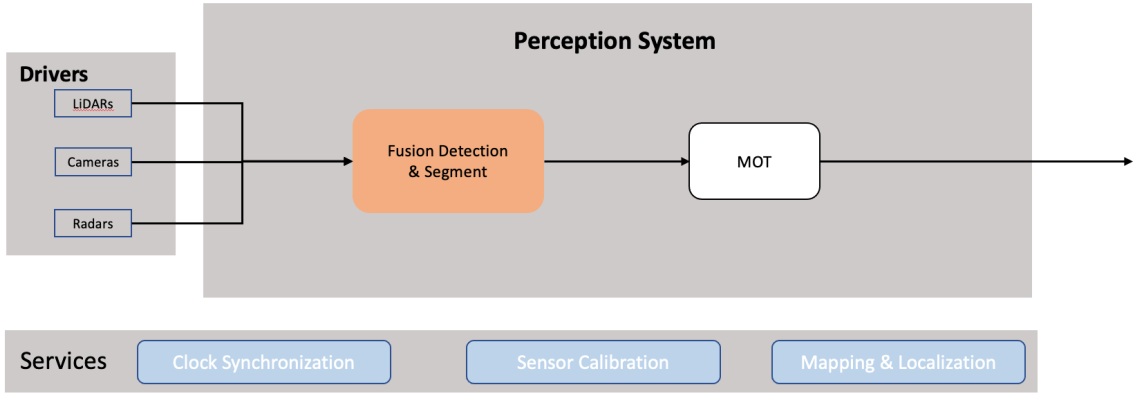
Camera+Radar融合算法：CRFNet

本章内容

1. 融合问题定义及背景
2. 后融合算法设计与实现
3. 前融合算法设计与实现
    1. DeepLearning及前融合介绍
    2. Camera/LiDAR/Radar之间的前融合
    3. 前融合与后融合的关系
4. 场景理解简介
5. 融合系统实现

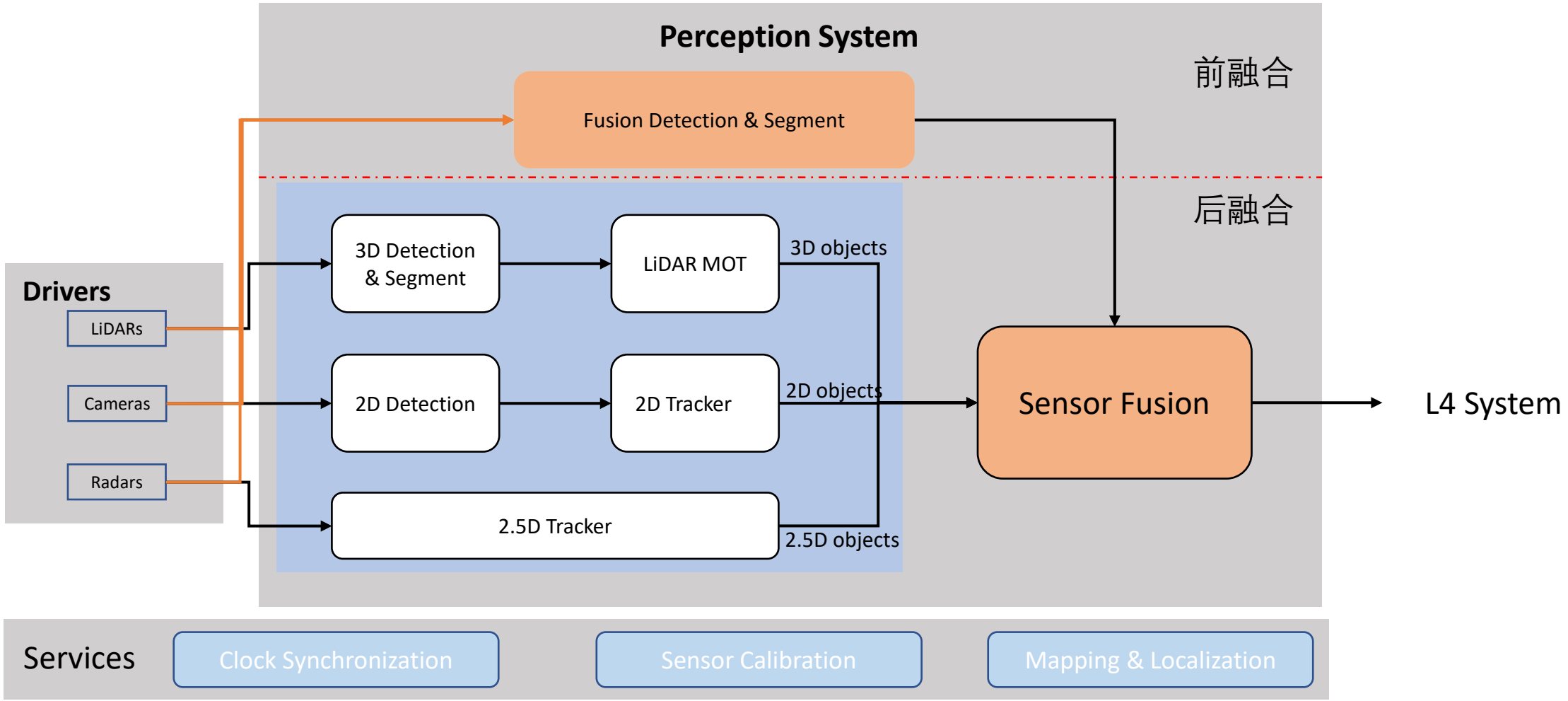
实际的多传感器融合的感知系统





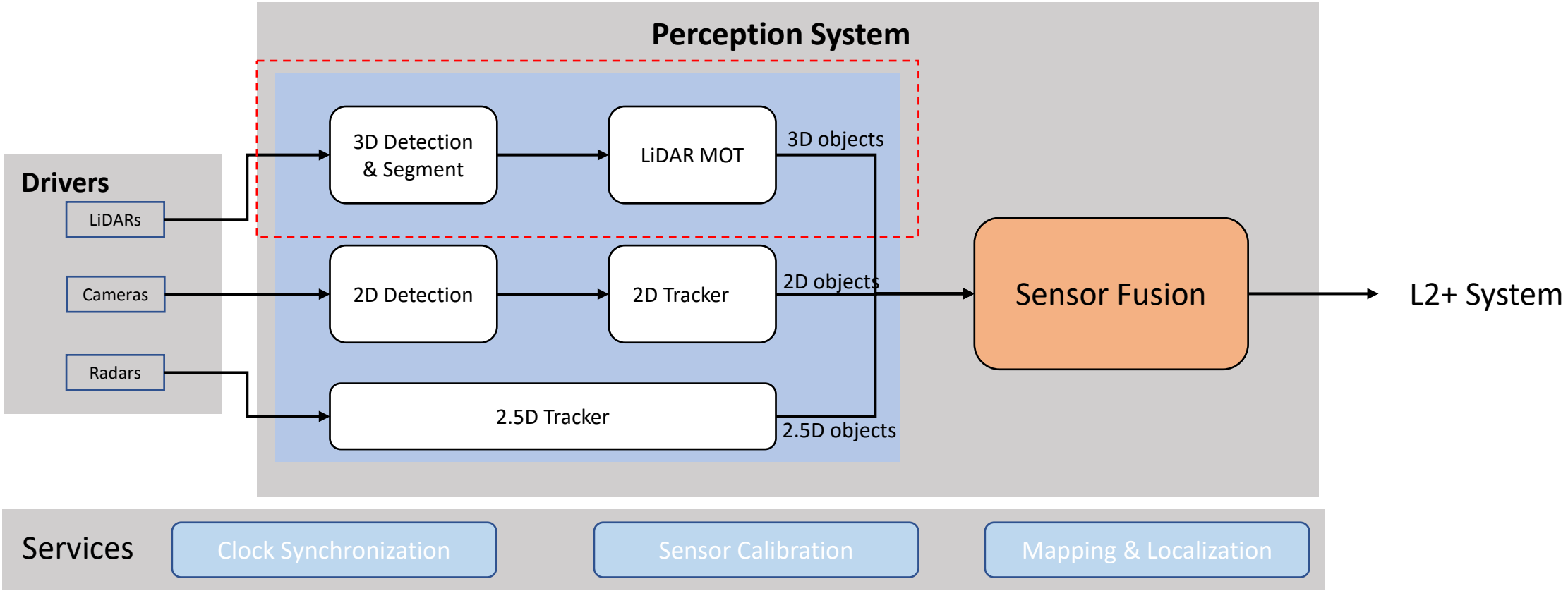
实际的多传感器融合的感知系统

通常现在的L4系统，都倾向于前后融合并行使用的模式。



实际的多传感器融合的感知系统

而L2系统，倾向于后融合模式。

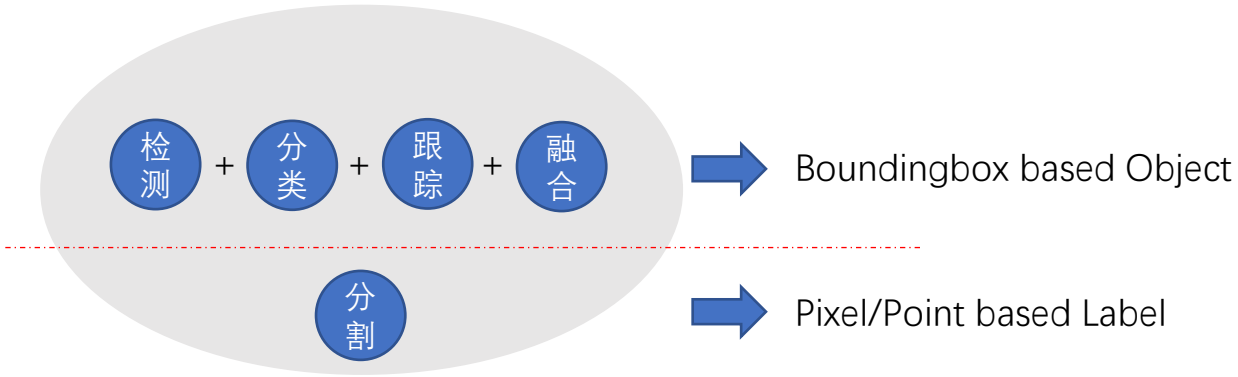


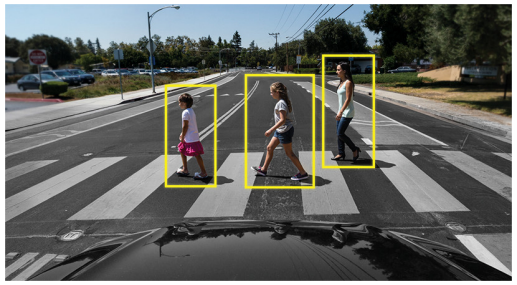
本章内容

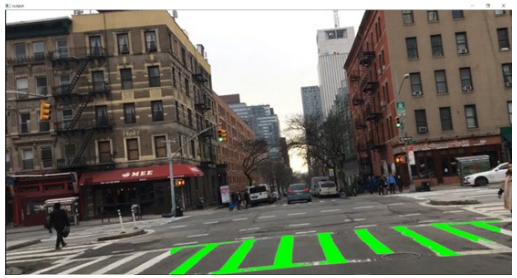
1. 融合问题定义及背景
2. 后融合算法设计与实现
3. 前融合算法设计与实现
4. 场景理解简介
5. 融合系统实现

场景理解：自动驾驶遇到的新问题

问题定义





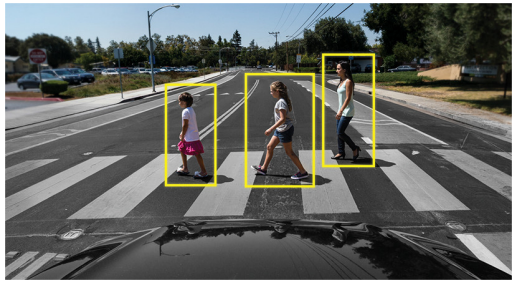


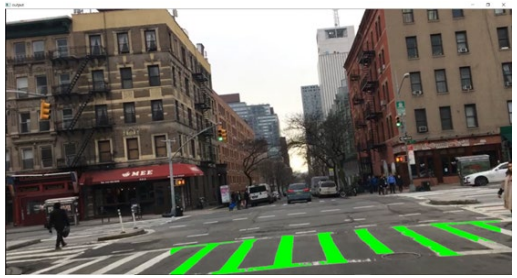
遇到一些由特定元素构成的场景，自车需要针对性更换形式状态，类别的进一步细化：

- 如何识别『斑马线上等红绿灯的行人』？
- 如何识别『施工场景』？
- …

基于规则的场景理解

如何识别『斑马线上等红绿灯的行人』？



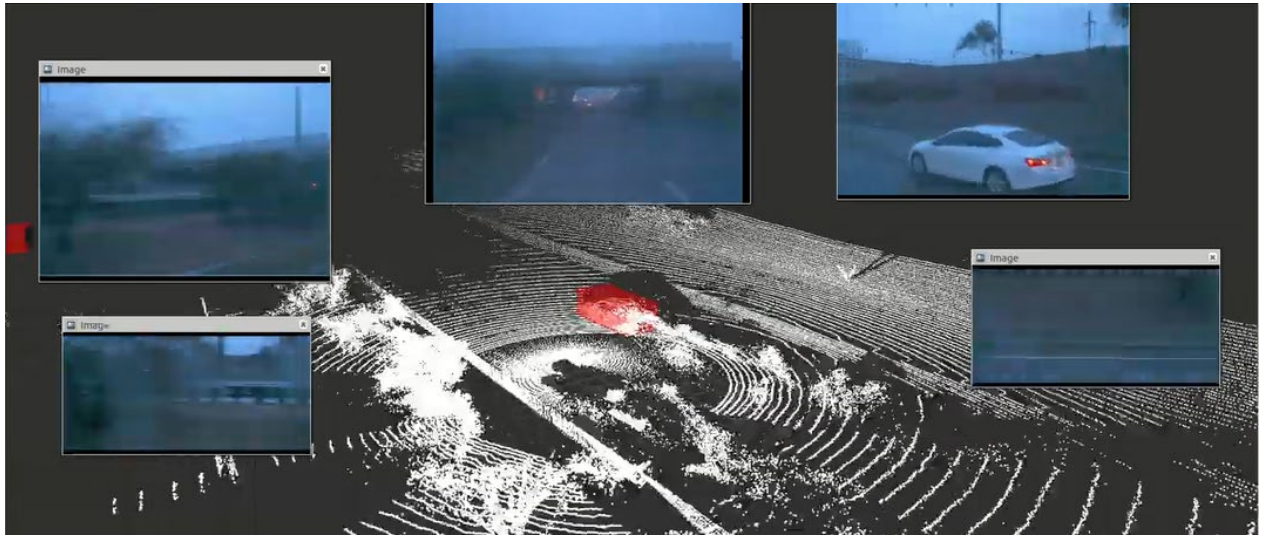


- 为什么需要细分？
  - 常规运动轨迹预测：可能会向各个角度产生运动；
  - 实际轨迹预测：当红灯时，按静止处理；当绿灯时，沿着斑马线匀速行走。

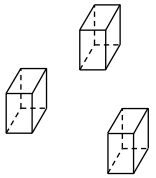
- 如何理解该场景？
  - 红绿灯场景
  - 人群运动特性：$\begin{cases}\Delta s < 0.3m \\ \Delta v < 0.5m/s\end{cases}$

场景理解——如何通过道路元素之间的关联，重新理解3D空间

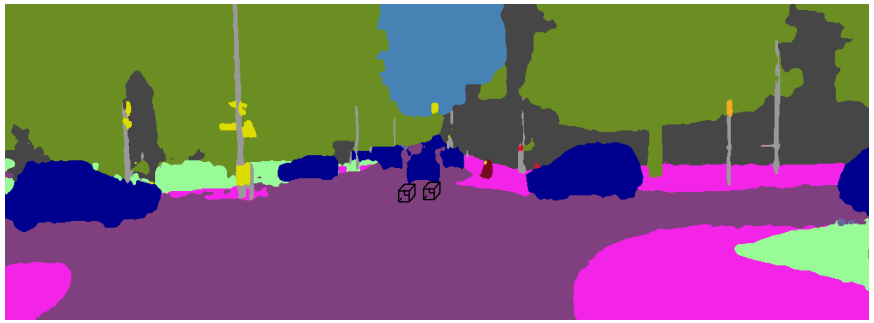
- LiDAR的雨水误检问题

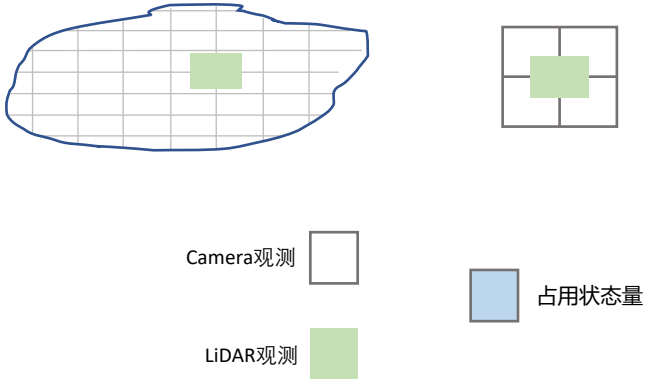


地面上没有东西



地面上有东西



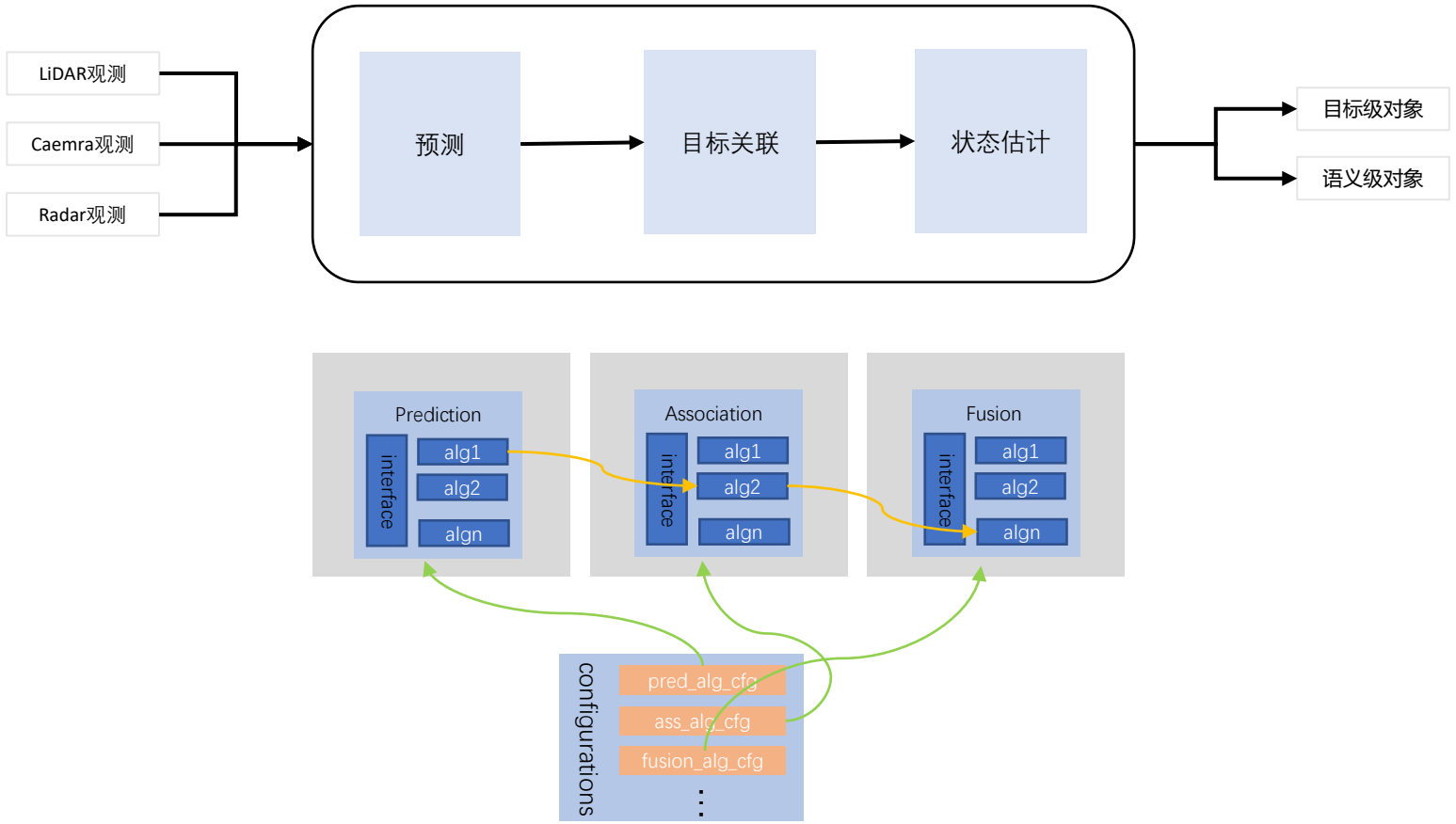


构建状态估计问题，用Kalman Filter可以解决上述问题。

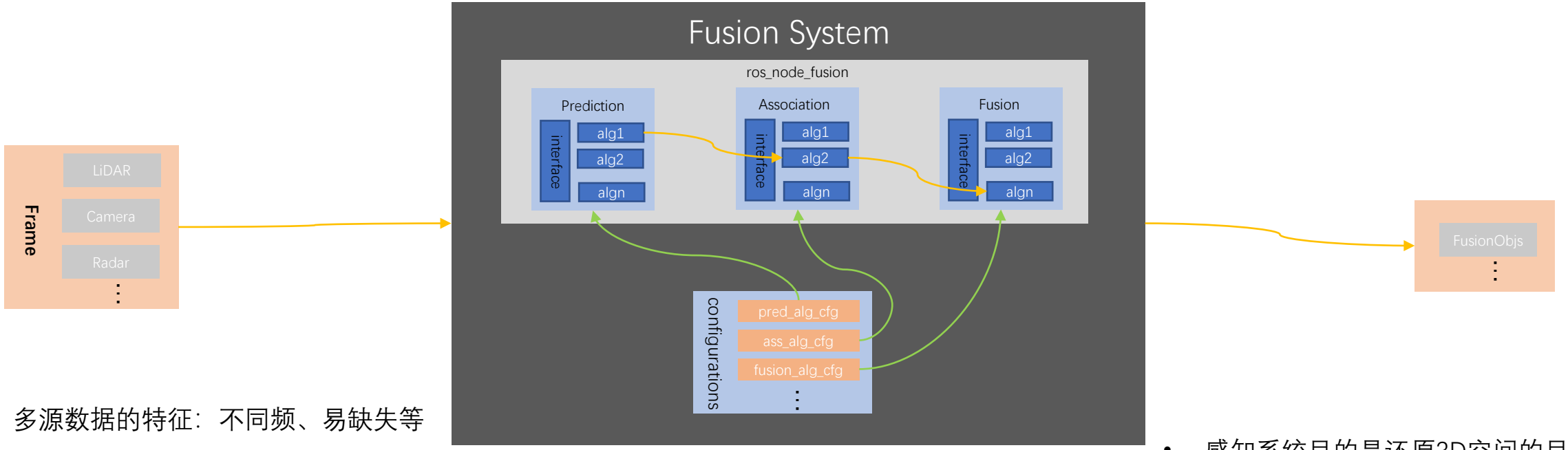
本章内容

1. 融合问题定义及背景
2. 后融合算法设计与实现
3. 前融合算法设计与实现
4. 场景理解简介
5. 融合系统实现

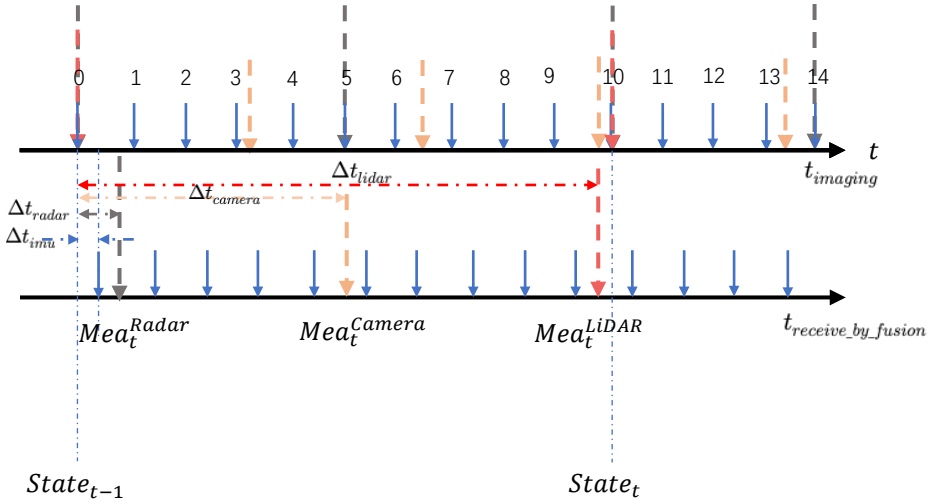
融合系统结构


LiDAR观测
Caemra观测
Radar观测
预测
目标关联
状态估计
目标级对象
语义级对象
Prediction
interface
alg1
alg2
algn
Association
interface
alg1
alg2
algn
Fusion
interface
alg1
alg2
algn
configurations
pred_alg_cfg
ass_alg_cfg
fusion_alg_cfg
...

融合系统设计



- 多源数据的特征：不同频、易缺失等



核心设计

- 主从传感器设计
  - 主传感器可创建状态量中的目标，主从传感器能更新状态量
  - 主传感器有备份：通常为LiDAR或Camera；
  - 从传感器：通常L4系统中的Camera和L2系统中的Radar
  - 指定周期内，主传感器数据未参与融合，则诱发系统中断。
- 融合触发模式
  - 主传感器触发：
    - 主传感器数据到达时，才触发融合，过程中的从传感器数据丢弃；
  - 所有传感器触发
    - 任何模态数据更新，立即触发多传感器融合动作；

- 感知系统目的是还原3D空间的目标，因此传感器本身的可信度不同，感知的目标置信度也不同，可用性：

$$LiDAR > Camera > \text{Radar}$$

深蓝学院
shenlanxueyuan.com

感谢聆听！

Thanks for Listening