

知
识
点

敲黑板，本文需要学习的知识点有

轨迹 障碍物

视觉感知 边缘分割

卷积神经网络 激光雷达

感知是自动驾驶的第一环，是车辆和环境交互的纽带。一个自动驾驶系统的整体上表现好坏，很大程度上都取决于感知系统的做的好坏。

讲到**感知**，首先不得不讲到**传感器**。**传感器**是自动驾驶感知环节中最主要的工具，我们必需对传感器能够提供的数据类型、适用的工况以及局限性都非常了解和熟悉，才能让算法工程师更好的适配传感器所采集和提供的数据写出更好的**算法**。

雷达感知是一种无线感知技术，通过分析接收到的目标回波特性，提取并发现目标的位置、形状、运动特性和运动轨迹，并且可以进一步推断目标和环境的特征。其作用类似于人类的眼睛和耳朵。

雷达感知可以支持非常丰富的应用场景，比如，毫米波雷达就已经被广泛应用在汽车辅助驾驶领域，用以检测行人和前车，实现防撞预警。此外，**雷达在居家、智能大厦、自动驾驶以及可穿戴设备**等领域也有很多潜在的应用。当前随着新型低功耗、小型雷达传感器的不断发展，雷达技术在很多智能设备以及电子产品当中都进行了很多应用。

以下，ENJOY

感知算法根据使用的传感器不同而不同。对于激光雷达Lidar点云感知，本节主要介绍两个检测算法，分别是**启发式的Ncut**和**深度学习算法CNNSeg**。对于视觉感知，将全面介绍深度学习检测和分割技术、跟踪技术，2D-to-3D技术以及多相机融合技术，此外还会介绍红绿灯检测算法。最后简要介绍Radar检测和超声波检测算法。

点云感知

点云障碍物感知的主要任务是感知障碍物的**位置、大小、类别、朝向、轨迹、速度**等。核心是**点云检测分割技术**，可以用启发式算法NCut和深度学习算法CNNSeg完成。

启发式方法：NCut

Ncut算法的基本思想是基于空间平滑性假设，即空间上接近的点来自同一个障碍物。首先，利用地图信息对点云进行预处理，例如去掉感兴趣区域之外的点云，降低点云图的复杂度。然后根据预处理后的点云构建加权图 $G=(V, E, W)$ ，将点云分割转换为图分割的问题，可以利用图聚类的算法求解，最终求解的每一个cluster就代表一个障碍物。该算法的优点是解释性好，缺点是分割规则过于简单，难以应对实际情况（草丛、绿化带），主要原因在于缺乏语义信息。

3

Part 3 感知算法：点云感知

- 启发式方法：NCut
 - 算法思路
 - 基于空间平滑性假设：空间上接近的点来自同一个障碍物
 - 基于点云构建Graph， $G=(V, E, W)$ ，
 - 把点云检测问题建模成Graph分割问题，一个cluster是一个障碍物
 - 预处理（ROI过滤，去地面）和后处理（异常过滤）
 - 基于Graph的聚类，是一个经典问题，也是经典谱聚类问题；有很多衍生算法；
 - 优点：解释性好；
 - 缺点：分割规则过于简单，难以应对实际情况；缺乏语义信息；

深度学习方法：CNNSeg

主要思想是利用**卷积神经网络**来处理激光雷达捕获的点云数据，并对点云中的目标进行识别。深度学习是数据驱动的，它把人工构造特征这一任务交给机器和算法去完成（即特征学习）。在百度的深度学习感知实现过程中，团队尝试了很多方法，下面就简单介绍算法研发的历程。

1、将所有点云都投到**前视图**（front-view）（投影面是一个圆柱面）来构造特征，将点云问题转化为矩阵问题，进而使用深度学习进行处理。通过构建全卷积神经网络对前向视图进行处理。

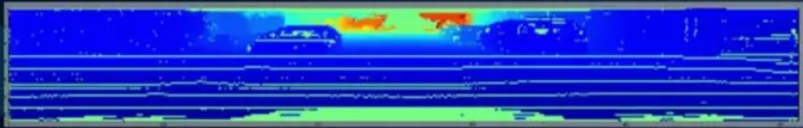
2、借助自采集车队，采集更多的实际数据，并且扩展数据视角，制作俯视图，通过将俯视图+前视图相结合的方式训练。同时，修改Loss函数，包括使用3D回归和Segmentation的损失函数。经过多次实验，发现基于**俯视图**的Segmentation方法效果最好。该方法目前已经在Apollo中开源，如下所示。

3


Part 3 感知算法：点云感知

- 深度学习方法：CNNSeg
 - 更多研发尝试，逐步走向实用：更多数据（自采）+ 更多View

Front View



Bird View



Loss 设计	BV	FV	BV+FV
回归3D框 (Detection)	SSD, DenseBox, Faster-RCNN	DenseBox (我们的论文工作)	DenseBox
分类每个点 (Segmentation)	SharpMask ✓	SharpMask	

SSD: <https://arxiv.org/abs/1512.02325>

DenseBox: <https://arxiv.org/abs/1509.04874>

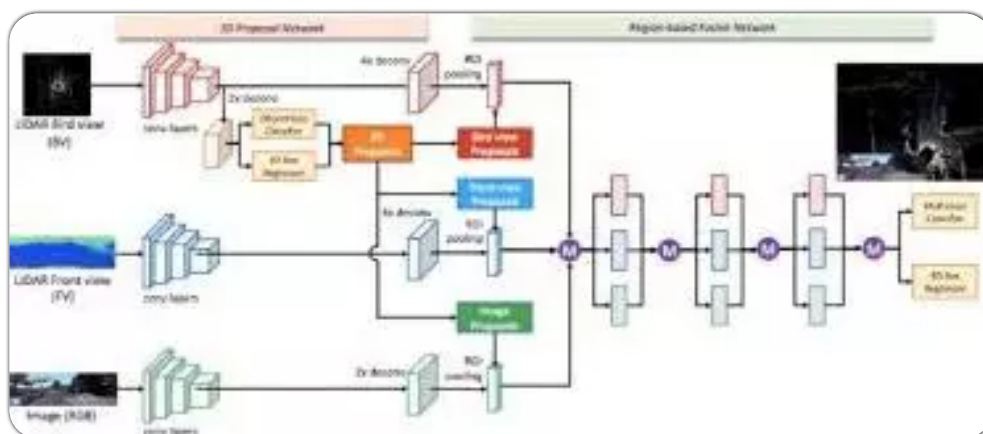
Faster-RCNN: <https://arxiv.org/abs/1506.01497>

SharpMask: <https://arxiv.org/abs/1603.08695>

▲ 基于俯视图的Segmentation方法

3、因为俯视图没有高度信息，因此我们把前视图和Camera图像加进来进行辅助检查，利用Lidar的测距准和Camera识别准优势完成了**Middle-Level Fusion方法**（Multi-View 3D Object

Detection Network for Autonomous Driving) , 发表在CVPR 2017。该方法使用俯视图提取 Proposal , 利用前视图和光学图像辅助进行更加精准的位置回归。其流程如下图所示。



▲ 俯视图提取Proposal的操作流程

视觉感知

视觉感知最早从ADAS发展而来。ADAS算法相对轻量，采用人工构造的特征和浅层分类器的方式实现辅助驾驶，该方法目前已经难满足自动驾驶的需求。随着深度学习技术的发展，尤其是在视觉领域的巨大成功，视觉感知的主流技术路线已经变为“**深度学习+后处理计算**”的方法。该方法带来了以下几个变化，第一是要求计算硬件升级，第二是数据的需求量大增，第三是如何评估保证安全性。面向自动驾驶的深度学习算法具有以下几个特点：

- 2D感知向3D感知渗透，模型输出更丰富（后处理需要的3D信息、跟踪信息、属性信息等都会放在CNN中进行学习）
- 环视能力构建（传统方法靠一个Camera完成前向检测、碰撞检测、车道线检测。无人驾驶需要环视）
- 感知+定位+地图紧密结合

CNN检测

CNN检测是深度学习里一个十分火热的应用，从最开始的AlexNet,VggNet到ResNet等，持续提高了ImageNet图像检测的精度，更多的细节和最新进展可以参考Kaiming He（何凯明）的系列工作。需要注意的是，目前发表的大部分关于检测的成果都是面向计算机视觉应用的，与自动驾驶领域的检测还有很大的区别。

首先，自动驾驶中，摄像头是安装在车上的，汽车行驶在结构化、规则化道路上，面向的场景更具体，有很多的几何约束可以用来辅助检测。

其次，自动驾驶中的检测模型需要输出的信息更多，包括障碍物的**尺寸、朝向**。同时自动驾驶还需要考虑时序性。我们称之为局部的End-to-End，检测、2D到3D转换、跟踪三步是自动驾驶视觉感知的组成，后面两步都由CNN来学习，减少人工干预。



▲ CNN检测流程

第三是多任务学习，网络结构适配。自动驾驶需要针对不同的障碍物特征（车道线，道路边界，定位元素）进行识别，如果分别由专用模型处理，整个处理流程太长，无法满足要求，因此需要做多任务的识别和网络结构的适配。

最后是属性识别，即除了障碍物级别的输出以外，还需了解速度类别朝向等问题，例如车尾灯状态，车门开闭状态。可以通过标注数据，由数据驱动的**深度学习**来自动学习这些更细的属性。

CNN分割

分割 (Segmentation) 与detection在本质上是一样的，是对一个目标的不同力度的刻画。分割是一种更细粒度刻画物体边界信息的检测方法，不再是画框，而是进行**边缘分割**。在感知中，对于不规则物体，往往需要进行分割处理。例如场景分割和可行驶区域感知。场景分割可以用于多传感器融合，例如对绿植进行分割，结合LiDAR点云，就可以测出绿植的类别和距离。可行驶区域则可以对一些非结构化道路的**路径规划**提供支持。在车道线感知中，应视具体情况使用分割或者检测方法。

后处理

一个完整的系统除了深度学习模型，还需要做一些后处理，后处理是直接针对下游模块，对后续的影响比较直接。在视觉感知中，后处理主要分为三个部分。

第一是2D-to-3D的几何计算。2D到3D的转换需要考虑的因素包括：

- 相机pose的影响
- 接地点
- 稳定性

第二是时序信息计算，主要是针对跟踪处理，需要注意以下几点：

- 对相机帧率和延时有要求，要求跟踪必须是一个轻量级的模块，因为检测已经占据大部分时间
- 充分利用检测模型的输出信息（特征、类别等）进行跟踪。
- 可以考虑轻量级Metric Learning

- 相机布局决定融合策略，要做好视野重叠。

红绿灯感知

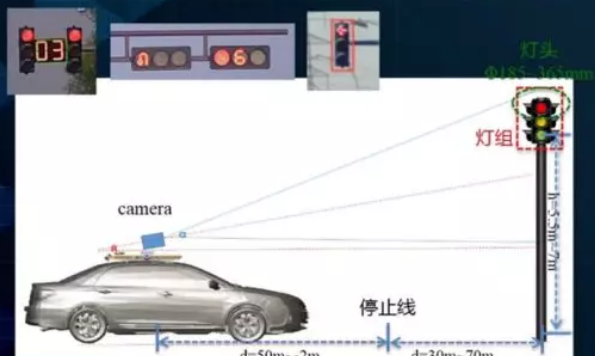
红绿灯感知是百度无人车第一个使用深度学习、使用Camera的模块。在当时使用深度学习、GPU和Camera做红绿灯感知是存在很多争议的，但是我们一直坚持走这种探索性的道路，并且成功了。现在车上用深度学习、GPU、Camera进行感知已经是一种共识。

红绿灯感知的任务是在距离停止线**50 ~ -2米**的范围内精准识别红绿灯亮灯状态。它的难点在于检测精度要求非常高，必须达到三个九（99.9%），否则会出现闯红灯，违法交规的情况。另外召回也不能太低，假设一直是绿灯，但是一秒只召回一帧绿灯，其他都认为是识别错的红灯，则通不过路口，影响通过率和体验。第二是红绿灯感知需要应对各种环境，包括天气和光照。最后是红绿灯的制式非常多，包括距离、高度、横纵、信号状态等，红绿灯感知都需要识别。

3

Part 3 感知算法：红绿灯

- 任务描述：距离停止线50~-2m，精准识别红绿灯亮灯状态
- 难点
 - 绿灯准确度高（3个9）否则闯灯，召回影响通过率和体验
 - 各种环境都work（光照和天气）
 - 各种制式红绿灯都可识别（距离，高度，横纵，信号形状，频闪）



▲ 红绿灯感知的任务和特点

Part 3 感知算法：红绿灯

- 技术途径
 - sensor选型和setup, 获取最大可视范围
 - High-Dynamic Range (HDR) 大于 100dB, 1080p
 - 双camera, 长短焦切换, 6mm + 25mm
 - 高精地图交互, 增加在线识别鲁棒性
 - 地图提供灯组3D坐标和交通含义
 - 只看RoI区域, 避免形状识别
 - 深度学习, 应对appearance变化
 - 检测: 灯组/灯头在哪?
 - 分类: 什么颜色?



▲ 红绿灯感知的技术途径

基于深度学习的红绿灯感知模块

自动驾驶中使用深度学习进行红绿灯感知模块的构建，主要分为以下几步。

第一是**相机选择和安装**。相机选择需要注意几个重要的参数，一是相机的高动态比HDR要高于100db。第二是拍摄图像的分辨率要达到1080P，满足15pixel大小的灯能够看得到。在实际应用中，我们使用了两个Camera，叫双Camera，长短焦切换，六毫米适用于近距离。

第二是**高精地图的交互**。我们并不是把所有红绿灯识别算法都交给在线算法完成，这样做无法满足鲁棒性要求也加重算法负担，需要把一些固定信息交给高精地图去完成，算法只实现在线变化的一些因素。地图提供灯组3D坐标和交通含义，算法只看RoI区域，避免形状识别出错。

最后使用深度学习识别灯颜色的变化。主要分为检测和颜色分类两步。

3

Part 3 感知算法：红绿灯

- 地图映射框(x3)：a,b,c,d
- 算法检测框：e,f,g,h,i,j,k
- Roi框：p



➤ 3D->2D投影

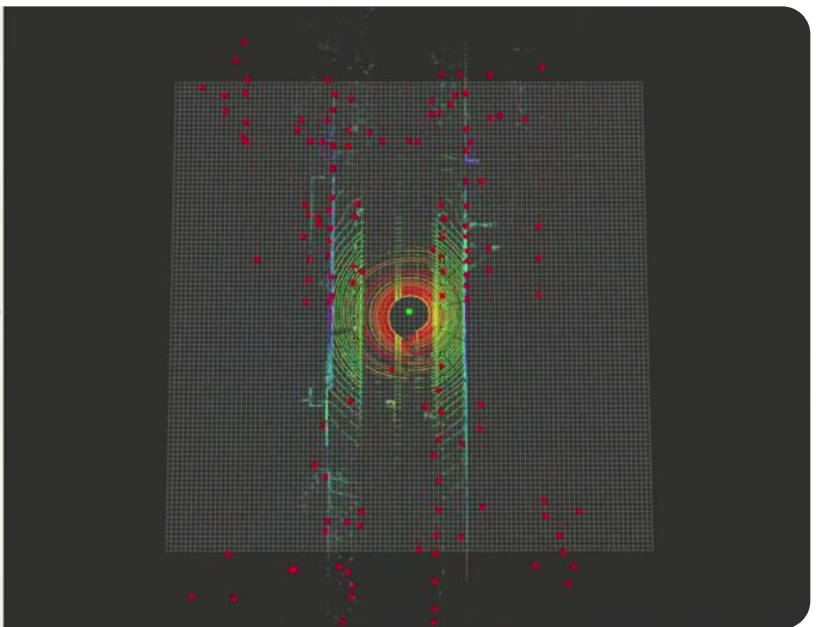
➤ 匹配算法

▲ 实际使用过程红绿灯感知

实际使用过程中可能会更复杂。如上图所示，一条道路上有很多红绿灯，检测算法会把所有的灯都检测出来，地图会告知需要看几个灯，但是并不知道看哪几个灯。因此需要把对应关系匹配起来，需要做3D到2D的投影，投影又受到标定、定位、同步、地图等因素的影响，需要综合考虑才能解决好这个问题。

IV

Radar感知



▲ Radar感知

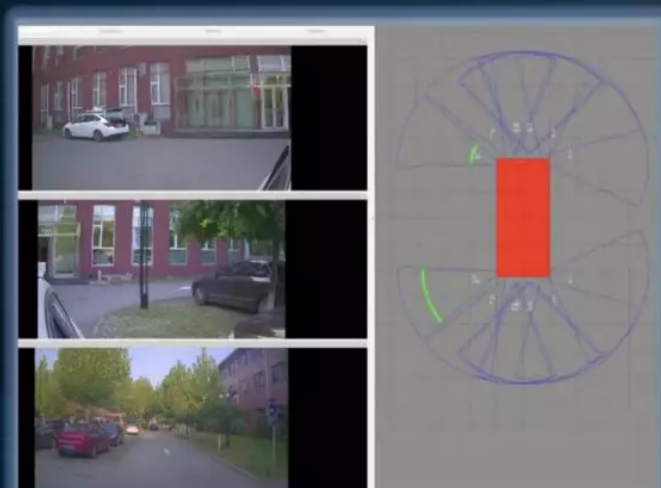
上图右边是Radar的点云信号投影的成像。图中有很多点，但是并不是所有的点都是障碍物，存在很大的噪音，需要进行鉴别。图中明显成一条线的点是对路边界金属栅栏的感知结果。因为金属的反射信号比较好，所以雷达点会成直线分布。基于该特征，可以用Radar反射信号来做高速路道路边缘栅栏的检测。

IV 超声波感知



3

Part 3 感知算法：Ultrasonic感知



▲ Ultrasonic感知

超声波只能进行近距离感知，并且无法感知具体的位置。对无人驾驶来说，行驶过程中帮助并不大，更多的是用于倒车和特别近距离的感知。上图给出了超声波感知的一个例子，我们在车上布了12个超声波传感器，每个超声波探头覆盖一个角度，大概检测到三米范围内的障碍物。当某个地方有障碍物，传感器发送超声波，收到回波信号以后可以得到障碍物的距离信息，但是并不能确定在覆盖范围内哪个地方有障碍物，如图中的绿线。如果两个探头感知范围存在over lap，这两个传感器的两根绿线会交接，就可以确定障碍物的准确位置。

END

