

自动驾驶综述|定位、感知、规划常见算法汇总

Self-Driving Cars: A Survey

自驾车自动驾驶系统的体系结构一般分为感知系统和决策系统。感知系统一般分为许多子系统，负责自动驾驶汽车定位、静态障碍物测绘、移动障碍物检测与跟踪、道路测绘、交通信号检测与识别等任务。决策系统通常被划分为许多子系统，负责诸如路径规划、路径规划、行为选择、运动规划和控制等任务。

一、自动驾驶汽车体系结构概述

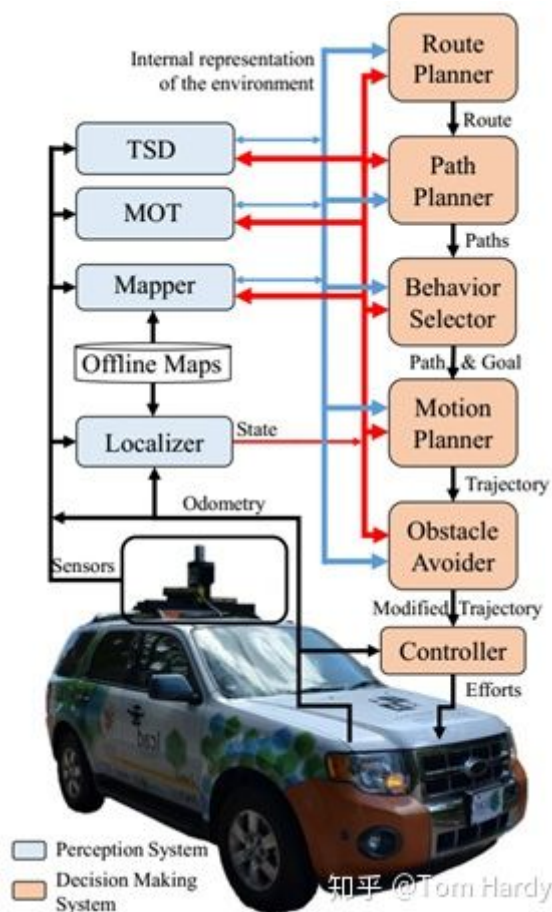
这一部分概述了自动驾驶汽车自动化系统的典型体系结构，并对感知系统、决策系统及其子系统的职责进行了评述。

下图显示了自动驾驶汽车系统的典型架构框图，其中感知和决策系统显示为不同颜色的模块集合。感知系统负责使用车载传感器捕获的数据，如光探测和测距（LIDAR）、无线电探测和测距（雷达）、摄像机、全球定位系统（GPS），惯性测量单元（IMU）、里程表，以及有关传感器模型、道路网络、交通规则、汽车动力学等的先验信息的决策。

决策系统负责将汽车从初始位置导航到用户定义的最终目标，考虑到车辆状态和环境的内部表现，以及交通规则和乘客的舒适度。为了在整个环境中导航汽车，决策系统需要知道汽车在其中的位置。定位器模块负责根据环境的静态地图估计车辆状态（姿态、线速度、角速度等）。这些静态地图在自动操作之前自动计算，通常使用自动驾驶汽车本身的传感器，尽管需要手动注释（即人行横道或红绿灯的位置）或编辑（即移除传感器捕获的非静态物体）。自动驾驶汽车可以使用一个或多个不同的离线地图，如占用网格地图、缓解地图或地标地图，进行定位。

定位模块接收离线地图、传感器数据和平台里程计作为输入，并生成自动驾驶汽车的状态作为输出。需要注意的是，虽然GPS可能有助于定位控制器的处理，但由于树木、建筑物、隧道等造成的干扰，使得GPS定位不可靠，仅GPS在城市环境中进行适当的定位是不够的。映射器模块接收离线地图和状态作为输入，并生成在线地图作为输出。该在线地图通常是离线地图中的信息和使用传感器数据和当前状态在线计算的占用网格地图的合并。在线地图最好只包含环境的静态表示，因为这可能有助于决策系统的某些模块的操作。为了允许检测和移除在线地图中的移动对象，通常使用移动对象跟踪模块或MOT。





行为选择器模块负责选择当前的驾驶行为，如车道保持、交叉口处理、红绿灯处理等。行为选择器根据当前驾驶行为选择目标，并在决策时间范围内避免与环境中的静态和移动障碍物发生碰撞。运动规划模块负责计算从当前车辆状态到当前目标的轨迹，该轨迹遵循行为选择器定义的路径，满足车辆的运动学和动力学约束，并为乘客提供舒适性。

二、感知模块

在这一部分中，我们研究了文献中提出的自动驾驶汽车感知系统的重要方法，包括定位（或定位）、离线障碍物映射、道路映射、移动障碍物跟踪和交通信号检测与识别。

定位模块负责估计自动驾驶汽车相对于地图或道路的姿态（位置和方向）（例如，由路缘或道路标记表示）。大多数通用定位子系统都是基于GPS的。然而，总的来说，它们不适用于城市自动驾驶汽车，因为GPS信号不能保证在封闭区域，如树下、城市峡谷（被大型建筑物包围的道路）或隧道中。文献中提出了各种不依赖GPS的定位方法。它们主要分为三类：基于激光雷达的、基于激光雷达加相机的和基于相机的。基于激光雷达的定位方法完全依赖于激光雷达传感器，具有测量精度高、处理方便等优点。然而，尽管激光雷达行业努力降低生产成本，但与相机相比，它仍然有很高的价格。在典型的基于LIDAR+camera的定位方法中，LIDAR数据仅用于建立地图，并使用相机数据估计自动驾驶汽车相对于地图的位置，从而降低了成本。基于摄像机的定位方法是廉价和方便的，尽管通常不太精确和可靠。

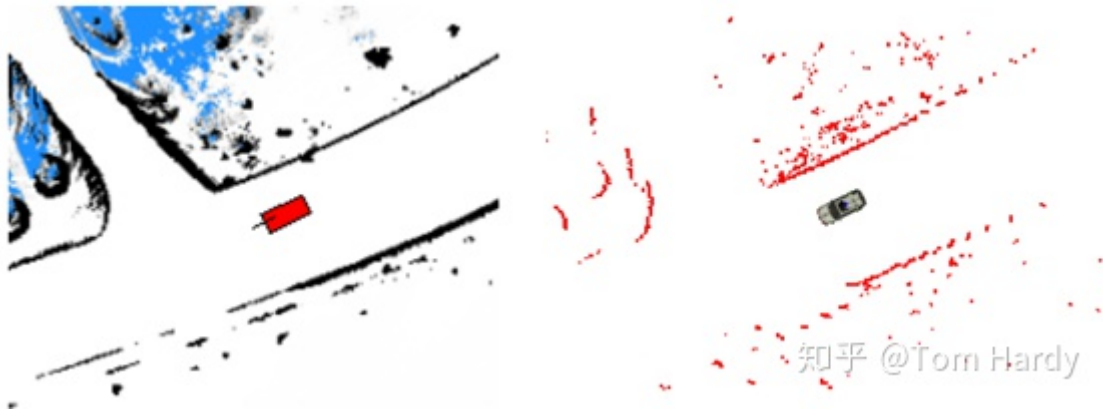


1、定位

1、基于激光雷达的定位

经典方法提出了一种结合三维点配准算法的多层自适应蒙特卡罗定位（ML-AMCL）方法。为了估计汽车姿态，从三维激光雷达测量中提取水平层，并使用单独的AMCL实例将层与使用三维点注册算法构建的三维点云地图的二维投影对齐。对于每个姿态估计，对一系列的里程测量进行一致性检查。将一致的姿态估计融合到最终的姿态估计中。该方法在实际数据上进行了评估，得到相对于GPS参考的位置估计误差为0.25m。然而，地图是昂贵的存储，因为它是一个三维地图。

Veronese等人提出了一种基于MCL算法的定位方法，该方法通过二维在线占有栅格地图和二维离线占有栅格地图之间的地图匹配来校正粒子的姿态，如下图所示。评估了两种地图匹配距离函数：改进了传统的两个栅格地图之间的似然场距离，以及两个高维向量之间的自适应标准余弦距离。对IARA自动驾驶汽车的实验评价表明，利用余弦距离函数，定位方法可以在100hz左右工作，横向和纵向误差分别为0.13m和0.26m。



2、激光雷达和相机方式定位

一些方法利用激光雷达数据建立地图，利用摄像机数据估计自动驾驶汽车相对于地图的位置。Xu等人提出了一种立体图像与三维点云地图匹配的定位方法。地图由一家地图公司（whatmms.com）生成，由几何数据（纬度、经度和海拔）和从里程表、RTK-GPS和2D激光雷达扫描仪获取的缓解数据组成。他们将地图的三维点从真实坐标系转换到摄像机坐标系，并从中提取深度和强度图像。采用MCL算法，通过将汽车摄像机拍摄的立体深度和强度图像与从3D点云地图中提取的深度和强度图像进行匹配来估计汽车的位置。该方法在实际数据上进行了评估，并给出了0.08 m到0.25 m之间的位置估计误差。VIS16提出了一种将地面全景图与一年中不同季节拍摄的卫星图像相匹配的自动驾驶汽车定位方法。在他们的方法中，激光雷达数据被分为地面/非地面类别。接下来，利用激光雷达数据将全景相机拍摄的自驾车地面图像分割成地面/非地面区域，然后进行扭曲以获得鸟瞰图。利用kmeans聚类将卫星图像分割成地面/非地面区域。然后利用MCL将鸟眼图像与卫星图像进行匹配，估计姿态。该方法在NavLab11自动驾驶汽车上进行了验证，获得了3m~4.8m的位置估计误差。

3、基于相机的定位方式



有些方法主要依靠摄像机数据来定位自驾车。Brubaker等人提出了一种基于视觉里程和道路地图的定位方法。他们使用OpenStreetMap，从中提取出感兴趣区域内连接他们的所有十字路口和所有可行驶道路（以分段线性段表示）。然后建立了一个基于图的路线图表示法和一个汽车如何通过该图的概率模型。利用这个概率模型和视觉里程测量，他们估计汽车相对于路线图的位移。使用递归贝叶斯滤波算法，通过利用图形的结构和车辆如何移动的模式（通过视觉里程计测量）在图形中执行推断。该算法能够通过增加当前姿势位于与最新汽车运动（直线行驶距离和最近的曲线）相关的图形点的概率，并通过降低其位于不相关的点的概率来精确定位汽车在图形中的位置。定位方法在KITTI视觉里程数据集上进行评估，并在行驶52秒后能够在包含2150公里可行驶道路的18平方公里地图上定位车辆，精度为4米。一些方法使用相机数据来构建特征地图。Ziegler等人描述了自主车辆Bertha在历史悠久的Bertha-Benz纪念路线上自动驾驶所使用的定位方法。提出了两种基于互补视觉的定位技术：基于点特征的定位（PFL）和基于车道特征的定位（LFL）。在PFL中，使用从当前相机图像中提取的DIRD描述符，将当前相机图像与先前在映射过程中获取的相机图像序列的图像进行比较。从映射过程中捕获的图像的全局位置恢复全局位置估计。在LFL中，地图半自动计算，提供道路标记特征（水平道路信号）的全局几何表示。通过检测从摄像机图像的鸟瞰图中提取的道路标记特征并将其与存储在地图中的水平道路信号相关联，将当前摄像机图像与地图匹配。然后，由PFL和LFL获得的位置估计被Kalman滤波器组合（作者不提供组合定位误差的估计）。Jo等人提出了类似于LFL的定位方法。一些方法使用相机数据来构造特征地图，但采用了其他类型的特征。Radwan等人提出了一种基于文本特征检测的定位方法。现成的文本提取技术用于识别环境中的文本标签。采用MCL算法对多个观测值进行融合。该方法在实际数据上进行了评估，并给出了1 m到25 m之间的位置估计误差。Spangenberg等人提出使用杆状地标作为主要特征，因为它们独特的、长期稳定的，并且可以被立体摄像机检测到。此外，它们允许内存高效的映射表示。特征检测主要由立体摄像机完成。定位采用MCL算法结合Kalman滤波器进行鲁棒性和传感器融合。该方法在自主车辆上进行了评估，得到了0.14m到0.19m之间的位置估计误差。一些方法建议使用神经网络对自动驾驶汽车进行定位。它们由相关的摄像机图像和相关的全局位置组成。在映射阶段，神经网络建立环境的表示。为此，它学习一系列图像和图像被捕获的全局位置，这些位置存储在一个神经地图中。在定位阶段，神经网络利用神经网络地图提供的先验知识来估计当前观测图像的全局位置。这些方法存在仪表刻度误差，难以实现大面积自主车辆的定位。

2、离线障碍物地图

离线障碍物地图子系统负责自动驾驶汽车环境中障碍物地图的计算。该子系统对于允许自主车辆在公共道路上安全行驶而不与障碍物（如路标、路缘）碰撞至关重要。障碍地图包含与汽车可能或可能无法导航的位置相关的信息，区分自由（可穿越）空间和占用空间。汽车必须总是在空余的地方。障碍物地图由地图绘制阶段的传感器数据构建，并存储在自主操作阶段供以后使用。状态空间的表示通常区别于拓扑和度量表示。拓扑表示将状态空间建模为一个图，其中节点表示重要位置（或特征），边表示它们之间的拓扑关系（例如位置、方向、邻近性和连接性）。这些分解的分辨率取决于环境的结构。度量表示通常将状态空间分解为规则间隔的单元。这种分解不依赖于特征的位置和形状。度量表示的空间分辨率往往高于拓扑表示的空间分辨率。这样的多功能性和高效性使得它们成为最常见的空间表现形式。

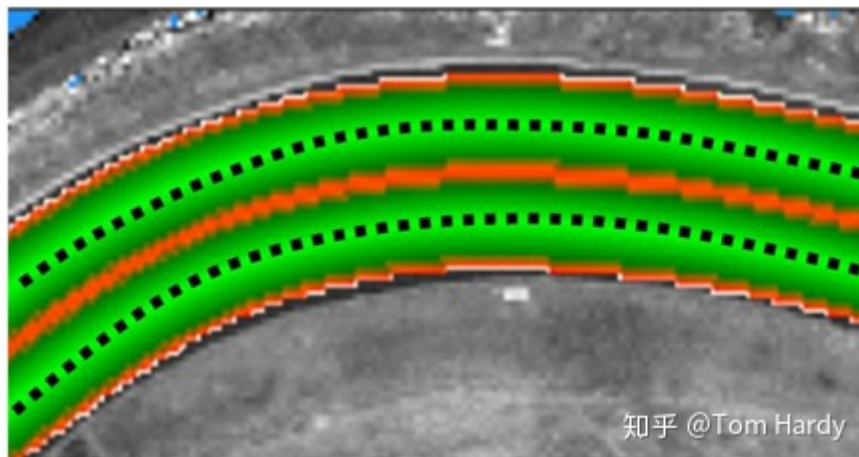


3、道路建模

道路地图子系统负责收集自驾车周围道路和车道的信息，并将其表示在具有几何和拓扑特性的地图中，包括相互连接和限制。道路制图子系统的主要内容是地图表示和地图创建。

1) 道路图表示与障碍地图一样，道路图通常区分为度量地图和拓扑地图。

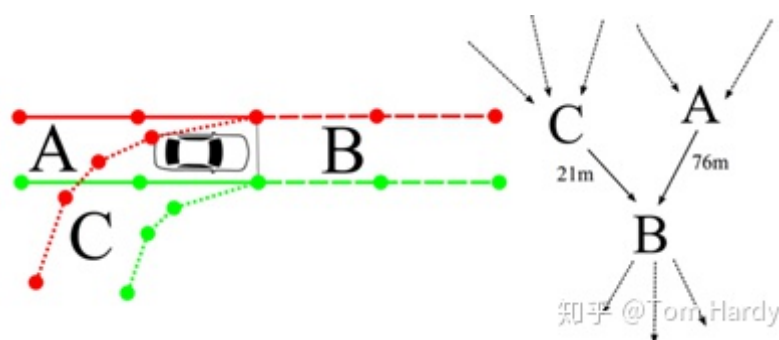
a) 度量表示路线图的简单度量表示是一个网格图，它将环境离散为一个固定大小的单元矩阵，该矩阵包含有关是否属于某条道路的信息以及移动到其相邻单元的成本。道路网格地图简单易懂。然而，如果移动成本在路线图的大范围内是一致的，那么使用网格表示可能需要浪费内存空间和处理时间。路线点序列是压缩大型道路网格地图中路径描述的一种替代方法。路线点是沿路线栅格地图中的路径的点。路线点序列可以手动定义，也可以自动从道路网格地图中提取。对于2005年DARPA大挑战，提出了路线数据定义文件（RDDF），它是一个格式化文件，包含指定自主车辆运行路径的航路点坐标和其他相关信息（纬度、经度、横向边界偏移和航速）。Carneiro等人为无人驾驶汽车IARA提出一个路线图，以推断城市道路中车道的位置和相关特性，该路线图同时使用了道路网格图和RDDF路径，如下图所示。IARA的道路网格地图包含 0.2×0.2 m的正方形单元格。为属于车道的每个单元格分配一个非零代码。从1到16的代码表示单元格到车道中心的相对距离，以及单元格中存在的车道标记类型（断开、实心或无）。IARA的RDDF路径包含间距为0.5米的航路点，并通过奖励靠近车道中心的单元格的算法自动从道路网格地图中提取。IARA的道路网格图和RDDF路径在联邦圣埃斯皮里托大学（UFES）主校区的环形道路上进行了3.7公里的自动测试。



b) 拓扑表示路线图的一种更复杂的表示是拓扑图，它将环境描述为一个图形模型，其中顶点表示位置，边表示它们之间的拓扑关系。拓扑图可以包含更复杂的信息，包括多车道、车道交叉口和车道合并。针对2007年DARPA城市挑战赛，提出了路线网络定义文件（RNDF），这是一个拓扑图，定义为指定无人驾驶汽车运行路段的格式化文件。根据该文件，道路网络包括一个或多个路段，每个路段包括一个或多个车道。路段的特征是车道数、街道名称和速度限制。车道的特征是车道的宽度、车道标线和一组航路点。车道之间的连接以出口和入口航路点为特征。厄姆森等人。URM08使用RNDF的图表模型作为自动驾驶汽车的老板（卡内基梅隆大学的汽车在2007年DARPA城市挑战赛中获得第一名）。图中的每个节点表示一个航路点，方向边缘表示将该节点连接到它可以到达的所有其他航路点的车道。基于多个因素的组合，将成本分配给边缘，这些因素包括穿过与边缘相关联的车道的预期时间、车道长度和环境的复杂性。Ramm等人。[RAM11]提出了



OpenStreetMap (OSM) , 它使用节点、方式和关系这三个基本体用拓扑图来建模环境。节点表示地理点, 方式表示节点列表 (多段线) , 关系由任意数量的成员组成, 这些成员可以是三种类型中的任何一种, 并且具有指定的角色。其他道路特性 (如行驶方向和车道数) 作为元素的特性给出。Bender等人。BEN14提出了一个高度详细的拓扑路线图, 称为lanelet地图, 用于自动车辆泊位。lanelet地图包括道路的几何和拓扑特征, 如道路、车道和交叉口, 使用原子互联的可驾驶路段, 称为lanelets, 如下图所示。lanelet的几何图形由左边界和右边界定义, 每个边界对应一个点列表 (多段线) 。此表示隐式定义每个车道的宽度和形状及其驾驶方向。lanelet的邻接构成一个加权有向图, 其中每个lanelet表示一个顶点, lanelet的长度表示其出边的权重。其他元素描述了限制条件, 如速度限制和交通规则, 如交叉口和合并权。lanelet地图在历史悠久的Bertha-Benz纪念路线上进行了103公里的自动测试。高清地图 (HD-maps) 是为无人驾驶汽车提供动力的新一代拓扑地图。高清地图具有厘米级的高精度, 包含丰富的信息, 如车道位置、道路边界和道路曲率。由于创建高清地图的成本很高, 因此有一些平台可以作为服务提供高清地图。Dharia对顶级供应商进行了评估和排名, 分别是Google、HERE、TomTom和Apple。



2) 路线图创建创建路线图的最简单方法是从航空图像中提取道路形状的手动注释。然而, 大型城市道路网所需的大量人工操作可能会使人工标注变得不可行。为此, 人们提出了从航空图像自动生成道路图的方法。

a) Urmson等人使用从航空图像中提取的道路形状的手动注释, 以便为自动驾驶汽车的驾驶台创建道路图。得到的局部道路形状是准确的, 但由于图像的分辨率和全局配准的影响, 全局位置不太准确。为此, 他们的定位方法采用位置滤波处理道路模型误差。Bender等人。BEN14还采用了自动车辆泊位的lanelet地图的所有元素和属性的手动标注。使用OSM格式和java OSM编辑器, 使用虚拟顶视图图像作为LANELET的手动注释的基础。

b) 自动生成从航空图像自动生成路线图的方法很多。韦格纳等人使用高阶条件随机场 (CRF) 通过将图像分割为超级像素并添加连接这些超级像素的路径来模拟道路网络的结构。Mnih和Hinton使用卷积神经网络 (CNN) 获得路段。道路分割的一个补充任务是从俯视图或正面图像中检测车道。Aeberhard等人对于宝马的自动驾驶汽车, 使用地面栅格地图, 其中每个单元表示具有高反射率的地面位置的概率。采用二次多项式模型提取道路边界。车道定位与数字地图结合使用, 以获得对环境的更高层次的理解。地图主要由两层组成: 语义几何层和定位层。语义几何层包含车道模型几何和车道连通性等高层语义信息。定位层包含车道标线和道路边界, 与GPS和车辆里程计一起, 可用于将车辆匹配到地图上。

Lee等人还使用激光雷达缓解数据来检测车道标记和摄像机图像，以防车道划分不明确。道路上的车道标记是为了在夜间与前照灯一起使用具有良好反光效果的特殊油漆而制成的。有了这个特性，激光雷达可以检测到道路标记，即使是在光照因雨或阴影而改变的情况下。基于摄像机图像的车道标线检测技术只在易受攻击的情况下运行（如背光和低光）。这种方法在韩国首尔2公里的航程中得到了成功的测试。Carneiro等人使用深度神经网络（DNN）来推断自主车辆IARA水平信号不良或无水平信号的车道的位置和相关特性。DNN将LIDAR缓解栅格地图分割为道路栅格地图，将非零代码（从1到16）分配给属于车道的地图单元，这些单元表示到车道中心的相对距离和单元中车道标记的类型。利用数十公里的道路标线数据集对DNN进行训练，使DNN的精度足以满足IARA的实际自动驾驶。道路分割并不直接提供路线图，它定义地图单元是否是道路的一部分。为了解释道路分割、提取拓扑结构和构建路线图，需要一个复杂的后处理流水线。巴斯塔尼等人提出了道路追踪方法，该方法寻求直接从CNN生成路线图，而不是依赖中间图像表示。它使用一个迭代的图形构建过程，一次添加一个单独的路段，并使用CNN来决定下一个要添加的路段。对15个城市24平方公里的航空影像进行的逐点匹配检验，平均误差为5%。

4、移动物体跟踪

运动目标跟踪（MOT）子系统（也称为多目标检测与跟踪-DATMO）负责检测和跟踪自动驾驶汽车周围环境中运动障碍物的姿态。该子系统对于使自主车辆做出决策和避免与潜在移动物体（如其他车辆和行人）碰撞至关重要。随着时间的推移，移动障碍物的位置通常是根据测距传感器（如激光雷达和雷达）或立体相机捕获的数据来估计的。单目摄像机的图像能够提供丰富的视觉信息，可以用来改进运动障碍假设。针对传感器测量的不确定性，采用Bayes滤波器（如Kalman和粒子滤波器）进行状态预测。MOT方法主要分为六类：传统的、基于模型的、基于立体视觉的、基于栅格地图的、基于传感器融合的和基于深度学习的。


1、Traditional Based MOT

传统的MOT方法主要包括三个步骤：数据分割、数据关联和过滤。在数据分割阶段，利用聚类或模式识别技术对传感器数据进行分割。在数据关联步骤中，使用数据关联技术将数据段与目标（移动障碍物）关联。在滤波阶段，对于每个目标，通过取分配给目标的数据的几何平均值来估计位置。位置估计通常由卡尔曼滤波或粒子滤波进行更新。

2、Model Based MOT

基于模型的方法直接从传感器数据中推断，使用传感器的物理模型和对象的几何模型，并使用非参数滤波器（如粒子滤波器）。不需要数据分割和关联步骤，因为几何对象模型将数据关联到目标。

3、Stereo Vision Based MOT

基于立体视觉的方法依靠立体图像对提供的颜色和深度信息来检测和跟踪环境中的运动障碍物。Ess等人提出了一种障碍物检测和识别方法，该方法仅使用来自前视立体摄像机的同步视频。

的工作重点是基于行人和汽车探测器每帧输出的障碍物跟踪。对于障碍物检测，他们采用了一种带有方向梯度直方图（HOG）特征的支持向量机（SVM）分类器，将每个图像区域分类为障碍物或非障碍物。对于障碍物跟踪，他们应用一种假设和验证策略，将一组轨迹拟合到可能检测到的障碍物上，使得这些轨迹一起具有很高的后验概率。候选轨迹集由扩展卡尔曼滤波器（EKF）生成，EKF由障碍物检测初始化。最后，使用模型选择技术仅保留一组解释过去和现在观测结果的最小且无冲突的轨迹。对于MOT，采用半全局匹配（SGM）方法从立体图像对中重构出稠密视差图像。三维环境中的所有障碍物都由一组称为超级像素或stixels的垂直方向的薄矩形来近似。使用Kalman滤波器跟踪随时间变化的stixel。最后，使用空间、形状和运动约束将stixel分割为静态背景和移动障碍物。在时空分析的基础上，提出了一种基于外观的检测与识别方案，该方案利用特定类别（行人和车辆）模型，提高了视觉感知的鲁棒性。

实时识别主要包括三个阶段：感兴趣区域（ROI）、障碍物分类和目标跟踪。Chen等人使用半全局匹配算法从立体图像对计算视差图。在视差图的辅助下，简单线性迭代聚类产生的图像分割边界分为共面边界、铰链边界和遮挡边界。利用改进的随机样本一致性（RANSAC）算法在自我运动估计过程中获得运动点。最后，根据边界类型和运动情况，采用超像素合并的方法提取运动障碍物。

4、Grid Map Based MOT

基于栅格地图的方法首先构建动态环境的占用栅格地图。地图构建步骤之后是数据分割、数据关联和过滤步骤，以便提供场景的对象级表示。Nguyen等人提出了一种基于网格的立体摄像机运动目标检测与跟踪方法。他们的工作重点是行人检测和跟踪。从立体图像对重建三维点。利用逆传感器模型，基于相关的三维点估计网格地图中每个单元的占用概率。采用分层分割的方法，根据网格单元之间的区域距离，将网格单元划分成若干段。最后，采用交互式多模型（IMM）方法对移动障碍物进行跟踪。Azim和Aycard使用基于八叉树的3D局部占用栅格地图，该地图将环境划分为占用、自由和未知体素。在构建局部网格地图后，基于局部网格地图中自由空间和占用空间的不一致性，可以检测出移动障碍物。动态体素被聚集成移动的物体，这些物体被进一步划分成层。使用从每个层提取的几何特征，将移动对象分类为已知类别（行人、自行车、汽车或公共汽车）。

5、Sensor Fusion Based MOT

基于传感器融合的方法融合来自各种传感器（如激光雷达、雷达和照相机）的数据，以探索它们各自的特点，提高环境感知能力。Darms等人介绍了自动驾驶汽车“Boss”采用的基于传感器融合的运动车辆检测与跟踪方法（卡内基梅隆大学的汽车在2007年DARPA城市挑战赛中获得第一名）。MOT子系统分为两层。传感器层从传感器数据中提取特征，这些特征可用于根据点模型或盒模型描述移动障碍物假设。传感器层还尝试将特征与来自融合层的当前预测假设相关联。无法与现有假设关联的功能用于生成新的建议。对与给定假设相关联的每个特征生成观察，封装更新假设状态估计所需的所有信息。融合层根据传感器层提供的建议和观测，为每个假设选择最佳跟踪模型，并使用卡尔曼滤波器估计（或更新）假设状态的估计。Cho等人描述卡内基梅隆大学新的实验性自主车辆使用的新MOT子系统。以前的MOT子系统，由Darms等人提出。Mertz等人使用可直接从二维激光雷达、从三维激光雷达投影到二维平面或从多个传感器（激光雷达、雷达和相机）融合获得的扫描线。扫描线被转换成世界坐标并被分割。为每个线段提取直线和角点特征。分段

有障碍物相关联，并使用卡尔曼滤波器更新目标的运动学。Byun等人合并由多个传感器（如雷达、二维激光雷达和三维激光雷达）生成的移动障碍物轨迹。将二维激光雷达数据投影到二维平面上，利用联合概率数据关联滤波器（JPDAF）跟踪运动障碍物。三维激光雷达数据被投影到一幅图像上，并使用区域增长算法分割成运动障碍物。最后，利用迭代最近点（ICP）匹配或基于图像的数据关联来估计或更新轨迹的姿态。Xu等人。XU15描述了卡内基梅隆大学的新型无人驾驶实验车对用于保持距离的移动障碍物的上下文感知跟踪。给定行为上下文，在道路网络中生成ROI。找到感兴趣区域内的候选目标并将其投影到道路坐标中。通过将来自不同传感器（激光雷达、雷达和摄像机）的所有候选目标关联起来，获得距离保持目标。薛等人融合激光雷达和摄像机数据，提高行人检测的准确性。他们利用行人高度的先验知识来减少错误检测。他们根据针孔摄像机方程，结合摄像机和激光雷达的测量来估计行人的高度。

6、Deep Learning Based MOT

基于深度学习的方法利用深度神经网络检测运动障碍物的位置和几何特征，并基于当前摄像机数据跟踪其未来状态。

5、交通信号检测与识别

交通信号检测识别子系统负责对交通规则中定义的标志进行检测和识别，使车辆能够根据交通规律做出正确的决策。与交通信号相关的任务有很多，在本文中，我们将探讨三个主要主题：交通灯、交通标志和自动驾驶汽车周围环境中的路面标记。

1、Traffic Light Detection and Recognition

红绿灯检测和识别包括检测汽车周围环境中一个或多个红绿灯的位置（例如，在图像中表示）并识别其状态（红色、绿色和黄色）。文献中提出了各种交通灯检测和识别方法。在这里，我们只回顾最新的和相关的。交通信号灯的检测与识别方法主要分为两类：基于模型的方法和基于学习的方法。交通灯在颜色和形状信息方面有一个明确的结构。一个普通的红绿灯有三个灯泡（每个州一个：红色、绿色和黄色）和一个清晰的形状。因此，早期的交通灯检测和识别方法大多是基于模型的。这些方法依赖于手工制作的特征工程，它试图利用人类掌握的关于物体颜色和形状的信息来建立一个能够检测和/或识别物体的模型。当假设没有严格遵守时，使用颜色和形状信息的方法是不可靠的。为了增强其鲁棒性，提出了一种不同特征（如颜色、形状和结构）的组合。例如提出了一个多特征系统，该系统结合了颜色（使用颜色分割）、形状/结构（使用黑盒检测）和地理信息（仅在预期有已知红绿灯时使用该系统）。然而，他们的系统在基于模型的方法中普遍存在大量的超参数，这通常意味着在某些情况下需要重新校准。作者在一个内部私有数据集上进行了实验，并指出故障是由于过度曝光、遮挡、交通灯的非标准安装以及其他一些在现实情况下并不罕见的情况造成的。在基于模型的方法的背景下，这种结合表明是不够的。因此，研究者开始引入基于学习的方法。在基于学习的方法中，特征仍然是手工制作的，但是检测和/或识别过程从基于规则变为基于学习。主要方法包括传统机器学习方式和深度学习方式。



2、Traffic Sign Detection and Recognition

交通标志检测与识别包括检测环境中交通标志的位置并识别其类别（如限速、停车和让行标志）。与基于模型的方法相比，基于学习的方法得到了改进并取得了更好的结果。随着深度学习在一般计算机视觉任务中的兴起，卷积神经网络已成为交通标志检测和识别领域的研究热点，在GTSRB和BTS的识别任务中分别达到了F1分数的99.71%和98.86%。

3、Pavement Marking Detection and Recognition

路面标线检测与识别包括检测路面标线的位置并识别其类型（如车道标线、道路标线、信息和人行横道）。路面标线检测与识别包括检测路面标线的位置并识别其类型（如车道标线、道路标线、信息和人行横道）。

大多数研究一次只处理一种路面标线，而不是同时处理所有标线。这可能是因为没有广泛使用的数据库，也没有一个共识，即研究人员在处理路面标线检测和识别时，应该关注哪些符号集。一个重要的路面标记是道路中的车道定义。早些时候，大多数车道标记检测方法都是基于模型或学习的。形状和颜色是最常见的特征，直线和曲线是最常见的车道表示。在BER17c中，作者提出了一个完整的自我车道分析系统。在这些系统的特点中，作者声称能够检测车道及其属性、人行横道、车道变换事件和一些路面标线。作者还发布了用于评估这些类型系统的数据集。深度学习是最近流行的另一种方法，像GUR16这样的方法已经显示出很好的效果。作者建议（i）使用两个横向安装的下向摄像机，（ii）将横向距离估计建模为一个分类问题，其中他们使用CNN来处理该任务。在这种情况下，他们主张在一个私人数据库中以小于2像素的平均绝对误差（MAE）达到亚厘米的精度。许多用于车道标记检测的方法也尝试用于道路标记检测。它们通常使用几何和光度特征。此外，各种道路标线检测与识别方法都采用了逆透视映射（IPM），降低了透视效果，从而使问题更容易求解，提高了结果的准确性。最近，一些方法使用最大稳定极值区域（MSER）来检测感兴趣区域（即可能包含道路标记的区域）和用于识别道路标记的卷积网络。在BAI17中，作者提出了IPM、MSER和DBSCAN相结合的算法来检测道路标线，并将PCANet与支持向量机或线性回归相结合进行分类。在单独评估分类任务时，它们的准确率达到99.1%，而在同时报告检测和识别性能时，其准确率则降至93.1%。在道路标记的上下文中，消息通常是单独处理的。一些消息检测和识别方法AHM17将不同的消息视为不同的类别（即，它们首先检测消息在场景中的位置，然后识别其类别），而大多数方法首先识别字母，然后使用基于OCR的方法进行写作。前者通常对天气和光照条件更为敏感，但后者可以识别看不见的信息。在道路标线的设置中，行人过街经常被单独调查。大多数人行横道检测方法利用人行横道通常呈现的规则形状和黑白图案。因此，在许多实际应用中，这项任务被放在了有利于鲁棒行人检测器的位置。

三、决策模块

在本节中，我们将对文献中所报道的自动驾驶汽车决策系统的相关技术进行研究，包括路线规划、行为选择、运动规划和控制子系统。



1、Route Planning

路线规划子系统负责计算从自驾车的初始位置到用户操作员定义的最终位置之间通过道路网络的路线。如果用一个加权有向图来表示道路网，其边权表示通过一个路段的代价，那么计算一条路线的问题就可以归结为在加权有向图中寻找最短路径的问题。然而，对于大型道路网络，经典的最短路径算法，如Dijkstra和A*的复杂度是不切实际的。在过去的十年中，道路网络中的路线规划算法的性能有了显著的进步。新开发的算法可以在毫秒或更短的时间内计算出行驶方向。道路网中的路径规划方法在查询时间、预处理时间、空间利用率和对输入变化的鲁棒性等方面提供了不同的权衡。它们主要可分为四类：goal-directed, separator-based, hierarchical, bounded-hop, and combinations。

1、Goal-Directed Techniques

目标导向技术通过避免扫描不在目标顶点方向上的顶点来引导从源顶点到目标顶点的搜索。A*是一种经典的目标导向最短路径算法。与Dijkstra算法相比，该算法在每个顶点上使用一个较低的距离函数，从而使更接近目标的顶点更早地被扫描，从而获得更好的性能。ALT（A*、地标和三角形不等式）算法通过选取一小组顶点作为地标来增强A*。在预处理阶段，计算所有地标和所有顶点之间的距离。在查询阶段，利用包含地标的三角形不等式估计任意顶点的有效下界距离。查询性能和正确性取决于是否明智地选择顶点作为标记。另一个目标定向算法是Arc Flags。在预处理阶段，图被划分成具有少量边界顶点和平衡（即类似）顶点的单元。通过从每个边界顶点向后生长最短路径树，为树的所有弧（或边）设置第i个标志，计算单元i的弧标志。在查询阶段，该算法将修剪没有为包含目标顶点的单元格设置标志的边。arc flags方法具有较高的预处理时间，但在目标定向技术中查询时间最快。

2、Separator-Based Techniques

基于分隔符的技术基于顶点或边分隔符。顶点（或边）分隔符是顶点（或边）的一个小子集，其移除将图分解为几个平衡的单元。基于顶点分隔符的算法使用顶点分隔符来计算覆盖图。快捷边将添加到覆盖图中，以便保留与完整图的任何顶点对之间的距离。覆盖图比完整图小得多，用于加速查询算法。HPML（High Performance multivel Routing，高性能多级路由）算法是这种方法的一个变种，它显著减少了查询时间，但代价是增加了空间使用量和预处理时间，在不同的级别上为图添加了更多的快捷方式。

基于弧分隔符的算法使用边界分隔符将图分解为平衡的单元，试图最小化连接不同单元边界顶点的切割边数。快捷方式将添加到覆盖图中，以保持每个单元内边界顶点之间的距离。

CRP（Customizable Route Planning，可定制路线规划）算法DEL15是为满足现实道路网络的需求而设计的，例如处理转弯成本和执行成本函数的快速更新。它的预处理有两个阶段。第一阶段计算多层分区和覆盖的拓扑。第二阶段通过自下而上和并行处理单元来计算团边的代价。查询作为覆盖图中的双向搜索进行处理。



3、 Hierarchical Techniques

层次技术利用了道路网络固有的层次结构，其中主要道路（如公路）组成一个小的干线子网。一旦源顶点和目标顶点距离较远，查询算法只扫描子网的顶点。预处理阶段根据实际的最短路径结构计算顶点或边的重要性。CH（compression Hierarchies）算法是一种分层技术，它实现了创建快捷方式以跳过重要性较低的顶点的思想。它重复执行顶点压缩操作，如果图中最短路径唯一且包含要压缩的顶点，则从图中删除最不重要的顶点并在每对相邻顶点之间创建快捷方式。CH是通用的，因此可以作为其他点到点算法和扩展查询的构建块。REACH算法是一种分层技术，在预处理阶段计算顶点的中心度度量（REACH值），并在查询阶段使用它们来修剪基于Dijkstra的双向搜索。设P是从源顶点s到包含顶点v的目标顶点t的最短路径。v相对于P的到达值是 $r(v, P) = \min\{\text{距离}(s, v), \text{距离}(v, t)\}$ 。

4、 Bounded-Hop Techniques

有界跳技术通过向图中添加虚拟快捷方式来预计算顶点之间的距离。由于预先计算所有顶点之间的距离对于大型网络是不可行的，因此有界跳技术的目标是在跳数很少的情况下获得任何虚拟路径的长度。

5、 算法的结合

可以将各种技术组合到利用不同图形特性的混合算法中。REAL algorithm结合了REACH和ALT。REACH Flags algorithm结合了REACH和Arc标志。SHARC算法[BAU09]将快捷方式的计算与多级弧标志结合起来。CHASE算法[BAU10]结合了CH和Arc标志。TNR+AF算法结合了TNR和Arc标志。PHAST算法可以与多种技术相结合，利用多核cpu和gpu的并行性来加速它们。巴斯特等人。使用众所周知的欧洲大陆大小的基准西欧和现实世界道路网络的简化模型，对这里描述的许多路线规划技术进行了实验性评估。

| Algorithm | Data Structures | | Queries | |
|--------------------|-----------------|------------|------------------|-----------|
| | Space (GB) | Time (h:m) | Scanned Vertices | Time (μs) |
| Dijkstra | 0.4 | — | 9,326,696 | 2,195,080 |
| Bidirect. Dijkstra | 0.4 | — | 4,914,804 | 1,205,660 |
| CRP | 0.9 | 1:00 | 2,766 | 1,650 |
| Arc Flags | 0.6 | 0:20 | 2,646 | 408 |
| CH | 0.4 | 0:05 | 280 | 110 |
| CHASE | 0.6 | 0:30 | 28 | 5.76 |
| HLC | 1.8 | 0:50 | — | 2.55 |
| TNR | 2.5 | 0:22 | — | 2.09 |
| TNR+AF | 5.4 | 1:24 | — | 0.70 |
| HL | 18.8 | 0:37 | — | 0.56 |
| HL-∞ | 17.7 | 60:00 | — | 0.25 |
| Table Lookup | 1,208,358 | 145:30 | — | 0.06 |

2、 Motion Planning



运动规划子系统负责计算从自动驾驶汽车的当前状态到行为选择子系统定义的下一个局部目标状态的路径或轨迹。该运动方案执行局部驾驶行为，满足汽车的运动学和动力学约束，为乘客提供舒适性，避免与环境中的静态和移动障碍物发生碰撞。

运动计划可以是路径或轨迹。路径是汽车状态的序列，它不定义汽车状态如何随时间演变。此任务可委托给其他子系统（如行为选择子系统）或速度剖面可定义为曲率和接近障碍物的函数。轨迹是一条指定汽车状态随时间演化的路径。

1、Path Planning

路径规划包括从汽车当前状态到下一个目标状态生成一系列状态，这并不定义汽车状态随时间的演变。路径规划通常分为全局和局部路径规划。在全局路径规划中，在车辆开始移动之前，使用环境的脱机全局地图计算全局路径。在局部路径规划中，当汽车行驶时，利用周围环境的在线局部地图生成局部路径，使汽车能够处理行驶中的障碍物。路径规划的方法主要分为两类：基于图搜索的和基于插值曲线的方法。

2、轨迹规划

轨迹规划包括从自动驾驶汽车的当前状态到下一个目标状态的序列生成，该序列指定汽车状态随时间的演变。轨迹规划方法主要分为四类：基于图搜索的、基于采样的、基于插值曲线的和基于数值优化的方法。

3、Control

在自动驾驶汽车领域，控制是指工程领域自动控制背后的理论，它涵盖了在没有持续的直接人为干预的情况下，应用各种机制来操作和调节过程。在最简单的自动控制类型中，控制子系统将过程的输出与期望的输入进行比较，并使用误差（过程的输出与期望的输入之间的差异）来改变过程的输入，从而使过程在受到干扰的情况下保持在其设定点。在自主车辆中，自动控制理论一般应用于路径跟踪和硬件驱动方法。路径跟踪方法的作用是在车辆模型存在误差等情况下稳定运动计划的执行。硬件驱动控制的作用是计算在执行器模型和其他模型不准确的情况下执行运动计划的转向、节气门和制动执行器输入。路径跟踪方法也被称为控制技术，因为它们采用自动控制理论，并将路径视为要控制的信号。然而，在自主车辆领域，将其称为路径跟踪方法更为合适，以区别于硬件驱动控制方法。

