# CMU-BOSS上的运动规划

首先，让我们来看看一张非常典型的图1-1无人车软件系统功能模块，从图中可以看到在行为决策层下游的模块就是动作规划（Motion Planning），下面我们主要来讲述在CMU-BOSS车上的动作规划系统。

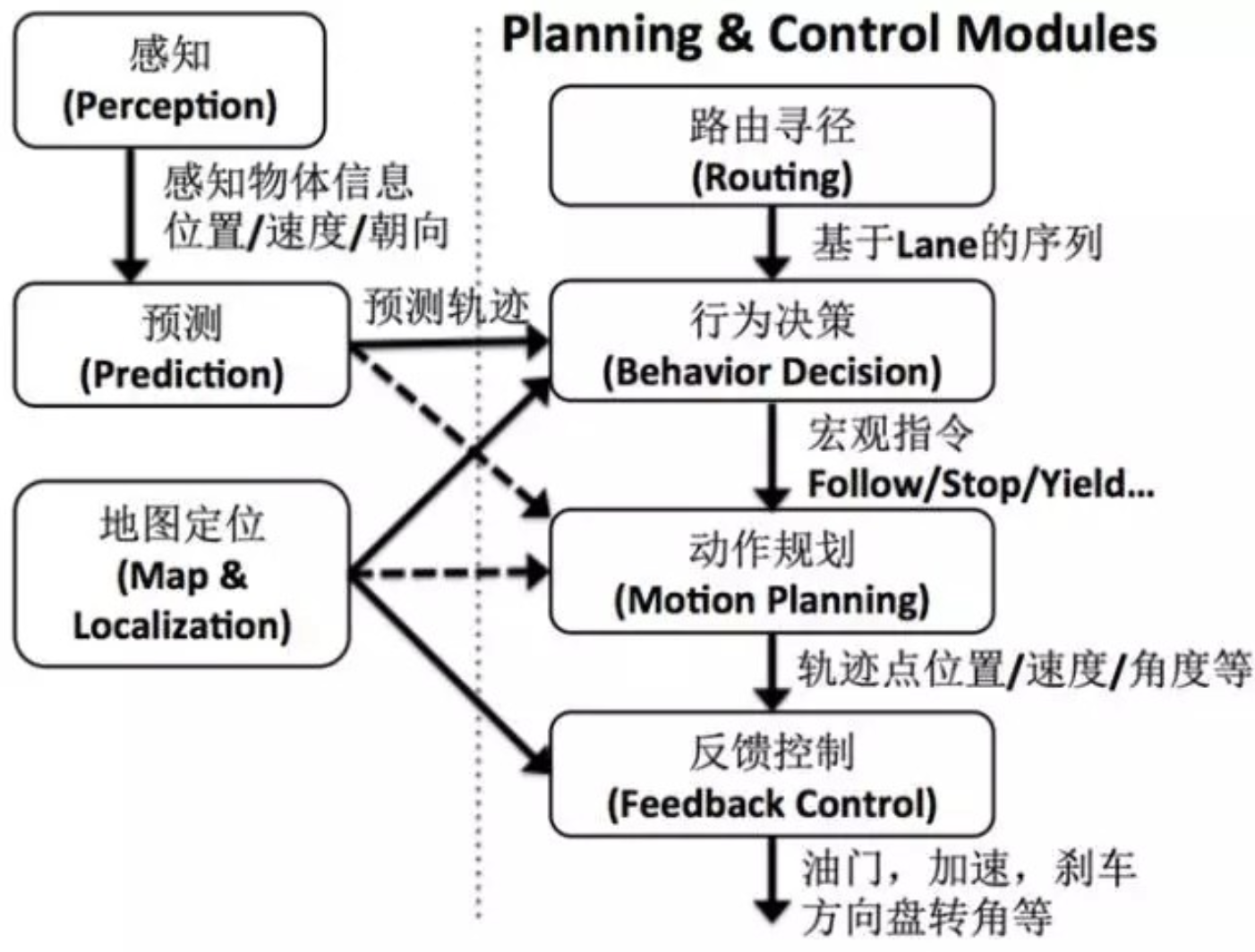


图1-1 无人车软件系统功能模块图

运动规划层负责处理从当前位置到行为层发出的目标位置的运动规划。运动规划层会考虑到周边的环境以及目标位置，从而确保车辆能正确地在道路上行驶。一般步骤为：1）得到运动目标 2）为该目标制定一个路径 3）生成一系列的候选轨迹，这些轨迹会随着移动而变化 4）评估函数选择出最佳轨迹（评估函数会根据当前周边的环境的不同而不同，考虑到的因素主要为一些静止的和移动的障碍物、路缘、速度、转弯曲率、是否偏移路径） 5）车辆按选定轨迹行驶。下面开始详细介绍各个步骤如何实现。

## 轨迹生成（Trajectory Generation）

最初，Howard和Kelly在2007年提出了模型-预测轨迹生成器（Model-predictive trajectory generator），这是最典型的候选轨迹生成方法，这个生成器能生成从初始状态X到目标状态的一系列可行操作。其中的\*算法\*可以解决一个参数化控制的生成问题**，**满足约束函数，其动力学能用一组微分方程来表示：。为了在城市环境中导航并且能够正确沿着道路行进，根据车辆位置和车子最终朝向，来确保车辆沿着道路。约束方程是约束函数C(x)与模型动力学积分的差值：

1. 2)

车辆模型的精确度直接关系到模型预测规划方法的有效性，车辆模型描述从控制输入到状态响应的映射(位置、方向、速度等)。选择一个合适的参数化控制是很重要的，因为它定义了满足边界状态约束优化的空间。

用于Boss的车辆模型结合了最小转弯半径、曲率速率极限(方向盘可以转动的最大速度的函数)、最大加速度和减速，以及控制输入延迟的模型。然后利用欧拉积分法对该模型进行模拟，来评估约束方程。控制输入由两个参数化函数描述: 基于时间的线性速度函数和基于弧长的曲率函数。线性速度图有4种形式，分别为恒定、线性上升、线性上升再恒定以及梯形速度（如图1-1）。局部运动规划器为特定的操作(如停车和距离保持)选择适当的参数化。

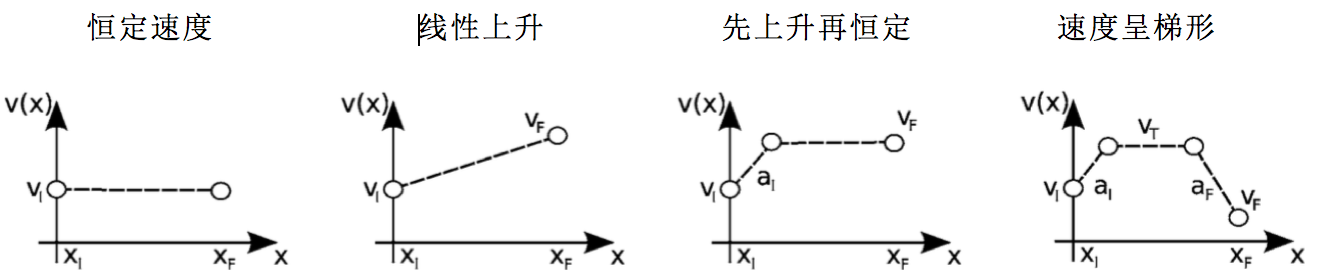


图1-2 基于时间的线性速度函数图

由车辆模型对曲率命令函数的响应定义了轨迹的形状。数据图表包括三个相关参数()(三个二阶结点数)和轨迹长度。在优化过程中，初始的结点数是固定的，以产生一个“smooth”或“sharp”轨迹，其中 ，这里可以看出系统保留这三个参数化自由。曲率指令函数使用弧长，因为轨迹形状与执行轨迹的速度无关。

考虑到系统的三个自由参数和三个约束条件，我们可以使用各种优化技术来求解最小化约束方程的参数值。在查找表中通过对状态空间到参数空间映射的预先计算，来定义一个参数的初始值。通过对描述运动方程的微分方程线性化和反转，参数估计值得以迭代地修正。修正数是通过取雅可比矩阵（雅可比矩阵是函数的一阶偏导数以一定方式排列成的矩阵）的逆矩阵和边界状态约束误差的乘积来得到的。因为雅可比矩阵是通过模拟车辆动作的中心差异来计算的，所以它是模型不变的。这些函数的偏导数(如果存在)可以组成一个 m行n列的矩阵，这个矩阵为，用符号表示为。

方程如下：

, ，

控制参数会一直修改，直到边界状态约束的剩余值在可接受范围内或者直到优化发散。如果给定一个特定的参数(例如，在最小转弯半径内)，边界状态约束是不可行的，那么优化将会出现分歧。所得到的轨迹作为最佳估计值返回，并由运动规划器评估。

## 道路导航（On-Road Navigation）

在道路行驶过程中，如果行为系统所给出的运动目标是车道内的位置，运动规划器负责成一条将车辆移动到目标所在的车道上的轨迹。为此，首先它需要在目标车道的中心线上构造一条曲线，车辆需要沿着这条线。随后这条曲线被转换为后轴坐标上的路径，将由运动规划器跟踪。

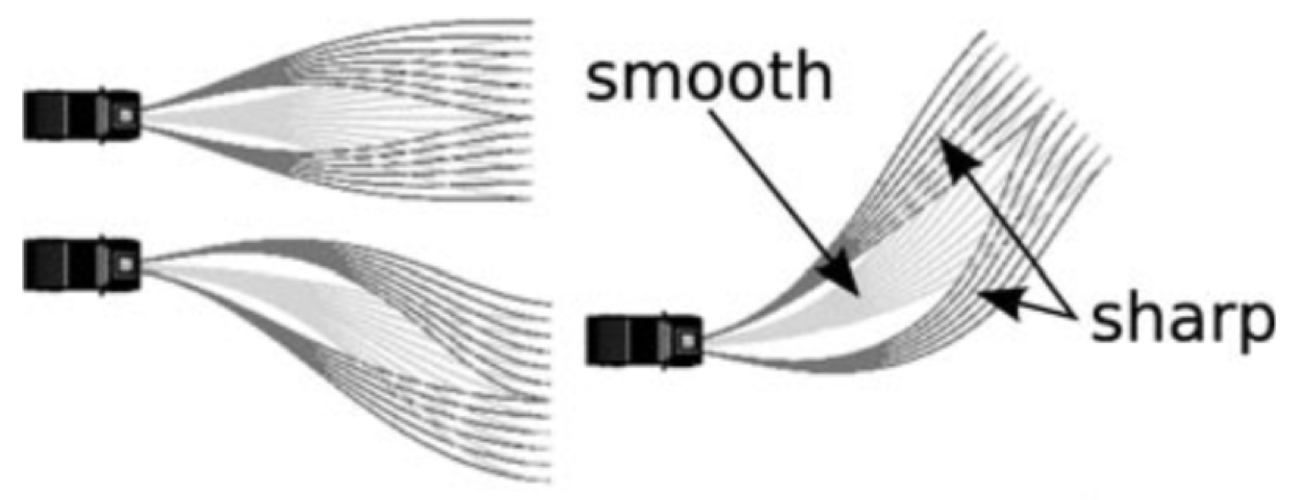
为了让车辆紧紧地跟随目标线路并且避免静止的和移动的障碍物，运动规划器产生的轨迹是一组来自中心线路径的局部位置点。（我们用S代表点在道路的中心线切向方向的位移，也就是纵向位移）。用L代表点在道路的中心线垂直方向位移，也就是横向位移）局部位置是中心线路径上恒定纵向距离的点（S相同），但它们在路径的侧向上偏移量不同（L不同），为规划器提供了几种选择。利用轨迹生成算法对这些局部位置动态计算出可行的轨迹，对于每个位置，产生两种轨迹: smooth轨迹与sharp轨迹。图1-2给出了smooth轨迹与sharp轨迹(黑色和灰色)。smooth轨迹有初始曲率参数，这个参数取决于车辆之前预测状态的曲率。sharp轨迹的初始曲率参数设置为一个偏移值，这个值来自前向预测的车辆状态的偏移值。 

图1-3 smooth轨迹与sharp轨迹

这些轨迹的行驶速度是基于几个因素，包括行为子系统所给定的最大速度、当前路段的限速、所给的路径曲率所能支持的最大可行速度和目标处所能允许的最大速度(例如,如果是停止线，那么速度就是零)。smooth轨迹表示出从头至尾有着连续的曲率控制。而sharp轨迹的曲率控制不连续并且跳跃，这就会导致车辆的剧烈晃动反应。

根据环境中的静止和移动障碍物、与中心线路径的距离、平滑度和各种其他各种指标来对产生的轨迹进行评估，选出来的最佳轨迹将由车辆执行的。轨迹生成器采用精确的车辆模型来计算每个轨迹的可行性，因此最后选择的轨迹可以直接由车辆控制器直接执行。

图1-3提供了一个在道路上行驶车辆进行轨迹规划的示例。(a)显示车辆行驶在双车道的道路上(蓝色表示车道边界，粉红色表示目前车辆的曲率，白色表示的最小转弯半径弧)。(b)显示了车辆所计算出的中心线路径（红色表示）。(c)用白色显示了一组分别距离红色中心线不同距离的候选路径，车子将从这组轨迹中，选择一个轨迹作为执行。(d)显示了根据环境中障碍物对轨迹进行评估。(e)红色表示中心线，蓝色表示已选定轨迹。(f)显示了车辆执行该选定的轨迹。

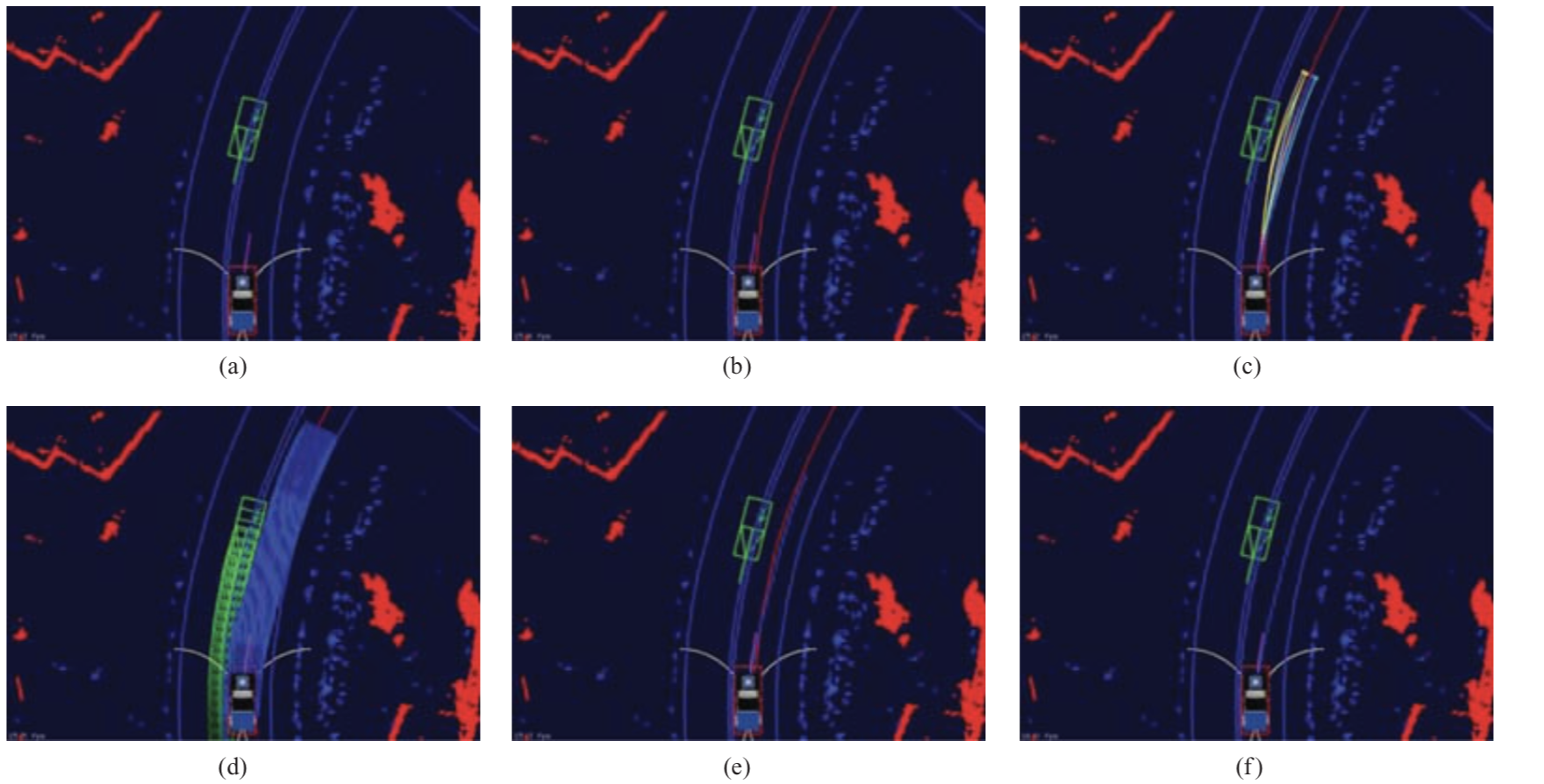


图1-4 拍摄下的规划图

## 区域导航（Zone Navigation）

在区域导航中，行为的运动目标是在一个区域内(比如一个停车位)中的某一个位置。运动规划器将产生一个运动轨迹，使车子朝着这个目标位置前进。然而，在非道路环境(如一大片区域，并且具有复杂障碍环境)中驾驶，这与在道路上行驶有很大的不同。如上一节所述，当在道路上行驶时，车道线(车道的中心线)潜在的成为了车辆的首选路径。所以在没有车道的区域内，车子的运动方式也就更自由了。

为了能有效地规划出一个能平稳到达远处目标的路径，我们使用栅格规划器（lattice planner）来实现，栅格规划可以对空间进行离散化，不依赖于具体的环境结构，而且构造方便，将车辆的位置坐标表示为(x, y)，方位为，速度为v。在动作规划器的搜索空间中，使用与轨迹生成使用所的车辆模型来构建每一个局部动作(x, y，,v)，模拟整个集合，这样车辆就可以准确按轨迹前行。运动规划器会在前进相反的方向进行搜索（目标位置 –> 区域位置），生成一条返回路径，这条路径是能够由车辆执行的，而且还规避了环境中的静止障碍物以及周边环境中一些存在碰撞危险的区域，例如在移动障碍物附近的一些区域。

使用栅格规划后，往往要使用图搜索算法来搜索最优轨迹，Dijkstra算法是经典的最短路径搜索算法。但是给定的一组初始位置、目标位置，由于采用广度优先方式将会导致太多无关节点的搜索，效率很低。A\* 是一种启发式最优搜索算法，其框架与 Dijkstra 基本一致，不同点在于引入了对当前结点到目标结点最低代价的估计函数，但对于环境复杂多变的情况，环境中的代价分布会发生变化，通过简单地重复使用A\* 虽然可以重新搜索出最优路径，但是效率较低。目前常用的解决方案是采用著名Anytime D\*算法(Likhachev, Ferguson, Gordon, Stentz, &Thrun, 2005)，该算法能根据新探测到的障碍物及环境信息及时修正候选可通行区域，缩小了候选可通行区域空间的搜索范围，速度较快。Anytime D∗可以快速地为车辆生成一个初步的、较优的路径，然后在时间允许的情况下完善这个方案。当接收到有关环境的新信息时(例如，观察到一个新的静止的或移动的障碍物)，Anytime D∗算法都能够根据新的信息来有效地校正目前解决方案。这种修复过程是通过逆向进行搜索（从目标位置到当前位置）来加快，因为在这种情况下，在车辆附近更新的信息只会影响搜索空间的较小的部分，从而更容易校正。

扩展到非常大的区域(最高0.5公里)，规划器使用了高精密搜索。在车辆以及目标位置的附近，就会需要更加复杂的机动，因此有着32个均匀间隔的方向来搜索这些区域。在其它区域，则只需16个均匀间隔方向来搜索。它还使用一组稀疏的动作，允许车辆在这些状态之间进行转换。由于粗糙搜索和密集搜索在16个朝向上是共同的，所以这两种精度切换能无缝地进行，产生的路径在空间的粗精度和高精度区域都是平滑可行的。

为了确保车辆在一进入区域有一个可行的路径，栅格规划器在车子还在接近这个区域的时候，就开始规划区域中某个点的路径。即使是在非常大的复杂区域，因为预先规划好从区域的入口点开始的路径，所以车辆还是可以连续的移动到目的地。在类似的情况下（复杂区域），当车辆行驶到一个停车点的区域时，第二个栅格规划器就计算从那个点到下一个预计位置的路径(例如，下一个停车点或离开区域)。当车辆到达预定的停车点时，车辆会立即跟随第二个栅格规划器规划的路线，再次消除等待路线生成的时间。

接着局部规划器以类似的方式跟踪所产生的计划，并在路线上获取这些路径。运动规划器生成一组轨迹，车辆尝试按照计划执行。还会生成反向，这个轨迹从车辆目前位置连接到路径上，每个轨迹实际上是两段短轨迹连在一起，第一段是车辆位置到偏移位置，第二段是偏移位置到路径。将所有的这种反向轨迹都连接到路径上，这样就显著降低了让车辆进入难以离开的状态的风险。

图1-5说明了栅格规划的跟踪能力和重新规划能力。这些图像是在NQE期间执行的一个停车任务中拍摄的 (左上角的图像显示了绿色区域和蓝色的周边道路)。右上角的图像中的彩色路径显示了车辆进入所指示的停车点（白色三角形）的初始路径，其他的一些区域被其他车辆占据(显示为不同颜色的矩形)，检测到的障碍物显示为红色。左下图显示出在移动过程中会动态调整轨迹(注意每个轨迹由两个部分组成，第一个离开路径，第二个返回路径)右下图显示出当车辆接近它的预定地点时，它观察到了停在最右边的停车点的车辆。到了这个位置，它意识到它的当前路径是不可行的，并且重新规划了一个新的路径，它使车辆绕一个圈才顺利地进入。这条路径在时间上更倾向于停止和后退以重新定位。

栅格规划器规划非常灵活，可以应对在道路和区域导航期间发生的各种情况。特别是，它可以在发生错误后的恢复过程中使用，比如说当在拥挤的中间路段航行、执行困难的U型转弯、驾驶演习中执行紧急操作后使车辆回到正常轨道。在这种情况下，行为层向运动规划器发出一个目标位置(或一组目标位置)，并让它处于错误恢复模式。然后，运动规划器使用栅格规划器来生成到目标位置的（一组目标位置）路径，在规划过程中，栅格规划器决定哪个目标是最容易达到的。在这些错误恢复场景中，栅格规划器会特意避免那些可能导致不安全行为的区域(比如在道路上出现的车道)。

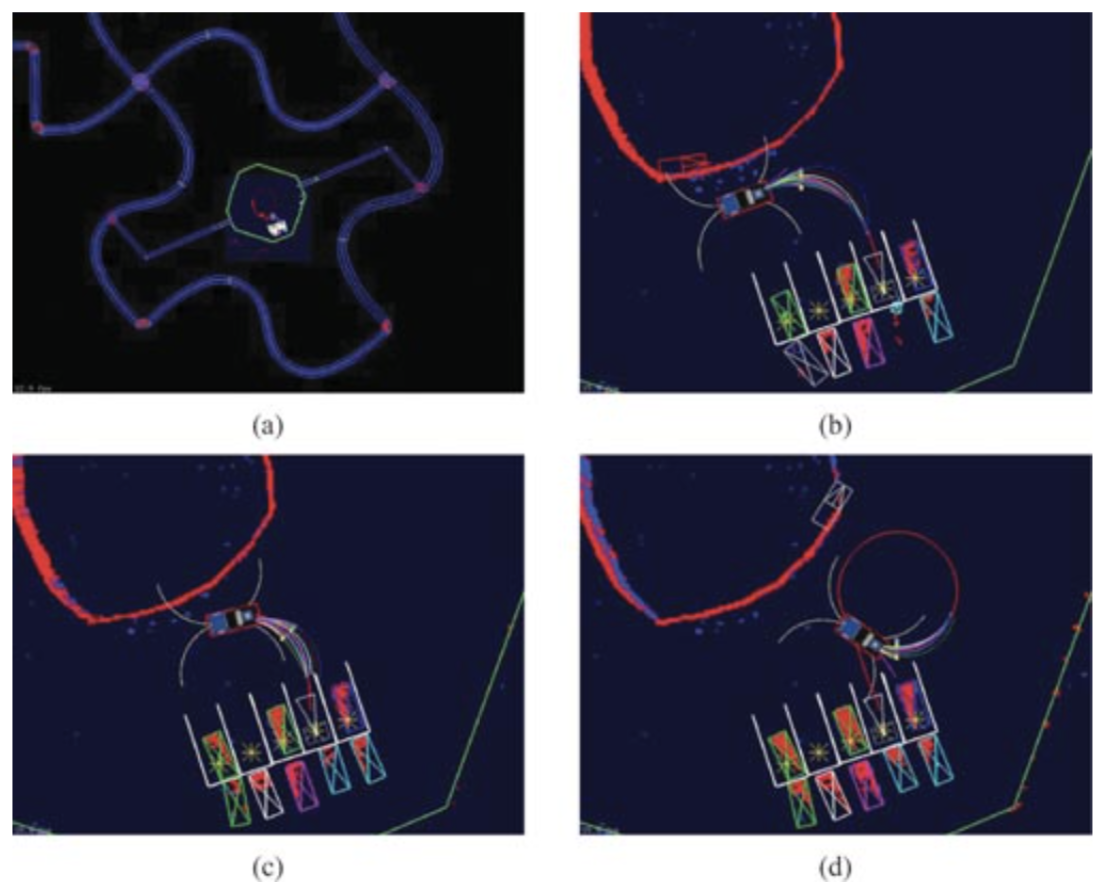


图1-5