

基于 AStar 算法的避障路径局部规划方法开发

刘志军, 苏亮, 吴少雄

(厦门金龙联合汽车工业有限公司, 福建 厦门 361023)

摘要:介绍一种基于 AStar 算法的避障规划逻辑,并介绍与之配合使用的路径平滑算法,再将 AStar 算法与实际场景相结合,提出一种多次规划的避障逻辑,最后将此逻辑与 Carla 进行联合仿真与调试。

关键词:路径规划; 避障; AStar 算法; Carla

中图分类号:U471.15

文献标志码:A

文章编号:1006-3331(2023)03-0006-04

Development of Local Planning Method for Obstacle

Avoidance Path Based on AStar Algorithm

LIU Zhijun, SU Liang, WU Shaoxiong

(Xiamen King Long United Automotive Industry Co., Ltd., Xiamen 361023, China)

Abstract: This paper introduces an obstacle avoidance planning logic based on the AStar algorithm, and presents the path smoothing algorithm used in coordination with it, then combines the AStar algorithm with the actual scene to propose a new obstacle avoidance logic of many times planning, and finally uses this logic and Carla to co-simulate and debug.

Key words: path planning; obstacle avoidance; AStar algorithm; Carla

汽车自动驾驶技术的路径规划可分为全局规划和局部规划^[1]。全局规划是指在一定的地图范围内规划出起点到终点的行驶路径,全局规划是一次性完成的;局部规划是指对全局规划结合实际环境信息进行规划路径的细化,当全局规划的行驶区域内出现障碍物时就需要局部规划来解决。而避障规划则是局部规划中非常重要的一环,能够大大提高自动驾驶的智能化程度。

一次性的全局规划通常先利用 AStar^[2] 进行路径点的搜索,再对结果进行平滑处理得到适合行驶的路径。而局部规划往往会因环境的变化,需要二次甚至多次规划才能得到合适的行驶路径。

本文基于 AStar 与路径平滑算法,结合实际应用场景,重新设计局部路径规划流程,实现用于避障的局部路径规划,最后利用 Carla^[3] 仿真环境进行模拟

与验证。

1 AStar 算法的介绍

AStar 算法是在 Dijkstra 算法基础上改进而来的一种启发式的路径搜索算法。而 Dijkstra^[4] 算法为基于广度优先空间搜索算法,从起始点周围的节点开始迭代扩展搜索至终点,从而得到起始点到终点的规划点信息。这种算法对算力要求高,且执行时间长。AStar 算法通过代价函数(式(1))^[5] 来确定迭代搜索的方向,避免盲目搜索导致的算力消耗,算力要求相对较低,且执行时间较短。

$$f(a) = g(a) + h(a) \quad (1)$$

式中: $f(a)$ 为节点 a 的代价; $g(a)$ 起始节点到节点 a 的代价; $h(a)$ 为节点 a 到目标节点的代价。

$h(a)$ 的计算可以是欧几里得距离 $h1(a)$,也可

收稿日期:2023-01-16。

第一作者:刘志军(1976—),男;主要从事产品研发管理、智能驾驶系统设计等工作。E-mail:liuzhijun@mail.king-long.com.cn。

通讯作者:吴少雄(1988—),男,高级工程师;主要从事自动驾驶车辆规划与控制算法的应用开发工作。E-mail:wusx@mail.king-long.com.cn。

以是曼哈顿距离 $h2(a)$ [6]:

$$h1(a) = \sqrt{(x_a - x_{goal})^2 + (y_a - y_{goal})^2} \quad (2)$$

$$h2(a) = |x_a - x_{goal}| + |y_a - y_{goal}| \quad (3)$$

式中: x_a 、 y_a 分别为节点 a 的 x 坐标和 y 坐标; x_{goal} 、 y_{goal} 分别为目标节点的 x 坐标和 y 坐标。

欧几里得距离 $h1(a)$ 计算的是节点 a 到目标节点的直线距, 需要进行平方与开根号计算, 会占用较多的浮点算力; 曼哈顿距离 $h2(a)$ 仅计算节点 a 得到目标节点横向与纵向的距离和, 只需要用到加法计算, 可以提高计算效率, 优先推荐使用。

AStar 算法会创建 $OpenList$ [2] 和 $CloseList$ [2] 2 个表对节点信息进行管理。其中 $OpenList$ 用来保存可能被搜索到的节点, 并根据代价函数计算的结果进行排序; $CloseList$ 用来保存已经被舍弃的节点, 算法不会对这部分节点进行搜索。当搜索完成后, $OpenList$ 将被清空。

2 路径平滑处理方法的介绍

本文方案的第一步是进行路径的平滑处理, 利用 AStar 算法最终得到离散路径点信息, 为了避免路径抖动带来的控制算法振荡问题, 需要使用贝塞尔曲线 [7] 算法对离散的路径点进行平滑处理。贝塞尔曲线有一阶、二阶和高阶之分, 一阶贝塞尔曲线为一条直线, 不满足使用要求; 高阶贝塞尔曲线则会消耗大量算力, 因而选择式 (4) 所示的二阶贝塞尔曲线进行路径平滑处理。

$$B(t) = (1-t)^2 P_0 + 2t(1-t) P_1 + t^2 P_2, t \in [0, 1] \quad (4)$$

式中: $B(t)$ 为 t 时刻下的坐标, 这里限定 t 在 1 s 内; P_0 为起始点坐标; P_1 、 P_2 为控制点坐标。

图 1 是使用二阶贝塞尔平滑前后的路径对比图。从图中可看出, 采用平滑算法后的路径曲线更加贴合实际车辆行驶的轨迹, 也更能让控制算法执行。

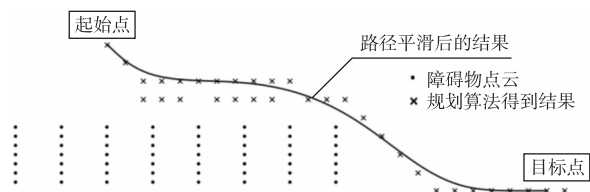


图 1 路径平滑算法

3 AStar 算法在绕障中的应用流程

智能车绕障场景是局部路径规划的一种典型应用, 如果要使用 AStar 算法进行规划则需要进行 2 次甚至多次规划。让其可以根据实时环境不断地调整规划路径, 根据此需求设计如图 2 所示的规划流程。

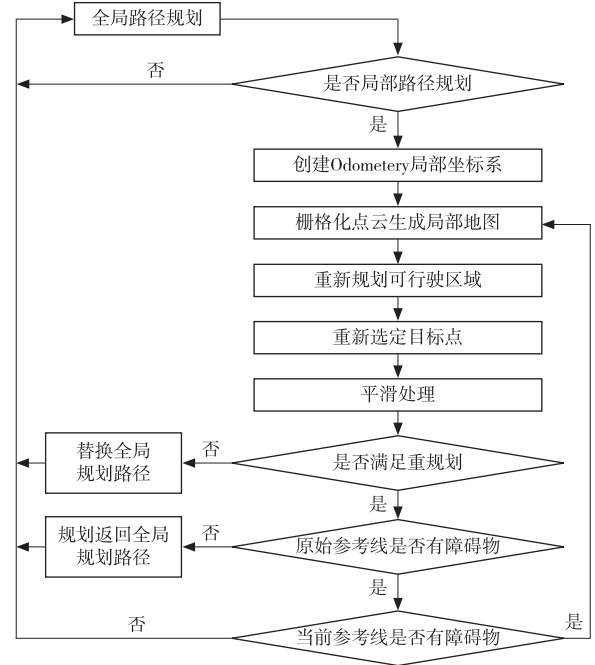


图 2 AStar 局部路径规划流程

如图 2 所示, 正常行驶情况的路径由全局规划负责, 当遇到障碍物并且满足避障条件时, 以车辆当前坐标为原点, 建立新的 $Odometry$ [8] 坐标系。整个绕障过程在这个新的坐标系中完成, 车辆完成绕障后, 驶回全局规划的路线上完成整个局部路径规划。路径规划前需要通过激光雷达获取车辆前方的点云信息, 并且生成 $costmap$ [9]。可行驶区域需要根据实际道路标线进行规划, 案例中规划的可行驶区域为单向双车道, 相比全局规划, 受限制的规划区域算法能够更快速完成求解。由于车辆前方有障碍物遮挡, 激光雷达无法探测障碍物后面的环境信息, 只能预估目标距离, 并在参考线上找到对应的参考点作为目标点。在 AStar 规划的结果上重新选取偏离参考线最远的点作为新的目标点, 将当前起始位置到目标点的路径通过贝塞尔曲线进行平滑处理, 处理后的曲线再与原有参考线进行拼接。当车辆运动到目标点时触发路径重规划, 如果原始参考线上没有障碍物, 则通过贝塞尔

曲线重新规划一条返回原始参考线的路径。如果当前参考线上仍然存在无法通过的障碍物,则通过 AStar 重新规划绕障路径。

4 AStar 绕障算法的可行性仿真分析

4.1 Carla 仿真平台简介

Carla 是一款支持自动驾驶系统开发、测试的开源仿真平台,本文采用该平台进行 AStar 绕障算法的可行性仿真分析。

要进行 AStar 绕障算法的仿真分析,首先需要进行相应的场景搭建,然后根据图 2 介绍的 AStar 局部规划流程编写自动驾驶程序,再通过自动驾驶程序进行车辆控制的仿真分析。

Carla 平台可以模拟多种环境下的城市道路、行人、车辆等原始要素,并将这些原始要素组合成开发者需要的仿真场景。同时 Carla 也支持多种传感器(GNSS、摄像头、激光雷达等),通过自带的 Ros-Bridge^[10] 功能实现 Carla 与 ROS Topic^[11] 或者其他第三方自动驾驶程序的交互,完成闭环仿真,如图 3 所示。

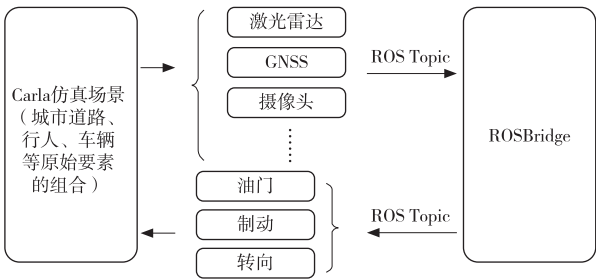


图 3 仿真环境与自动驾驶程序的交互

4.2 场景搭建

在 Carla 中构建一个用于模拟避障路径规划的场景,如图 4 所示。图中①为受控制的自主车辆(ego car),②为障碍车辆,线段③围成的区域为规划的可行行驶区域,线段④为全局规划得到的参考线。

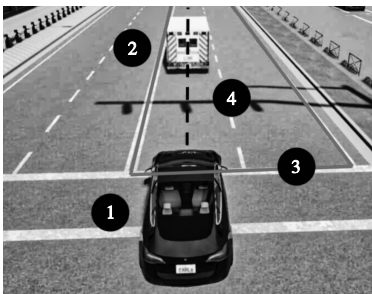


图 4 用 Carla 构成的避障场景

4.3 AStar 算法可行性仿真

自动驾驶程序可行性仿真分析包括地图生成、驶离全局规划、多次路径规划、返回全局规划。根据图 4 所搭建的场景运行自动驾驶程序,对绕障算法流程进行可行性验证。

4.3.1 生成局部地图

如图 5 所示,当前方障碍物侵入行驶路径造成无法通行,程序将栅格化激光雷达点云信息,并以当前位置作为原点生成局部地图。

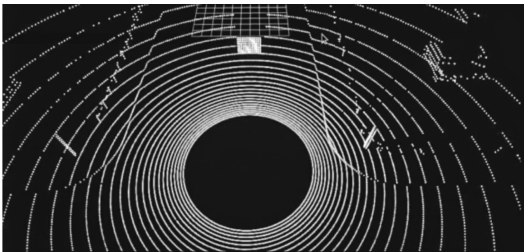


图 5 Rviz 中显示 Carla 环境数据

4.3.2 驶离全局规划路径

如图 6 所示,使用 4.3.1 节生成的局部地图信息,通过 AStar 算法得到一条能够避开前方障碍物的局部路径。



图 6 驶离全局规划

4.3.3 多次路径规划

由于激光雷达无法探测出前方障碍物的长度,当车辆通过 4.3.2 节的路径绕开障碍物后,激光雷达可以探测到车辆的长度信息。如图 7 所示,当前一次规划要走完时,仍然无法绕回原有路径,算法会继续规划,直到车辆完全绕开障碍物。

4.3.4 返回全局规划

如图 8 所示,经过多次路径规划,车辆已经完全绕开障碍物。当满足驾驶全局规划的条件时,算法会生成一条返回全局规划的过渡路径。

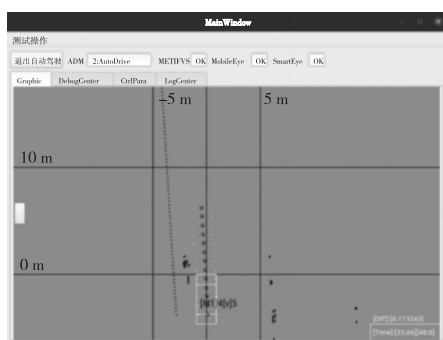


图7 多次规划



图8 返回全局规划

图6~图9为整个绕障算法的执行过程,均为自动驾驶程序视角。图6中车辆在全局规划的参考线上发现障碍物,规划出一条驶离全局规划的避障路径;图7为避障路径行驶结束后,自动驾驶程序发现原全局规划路径上仍然有障碍物而继续规划避障路径;图8为自动驾驶程序规划返回全局规划路径的过渡路径;最后在图9的回放中,可以看到整个避障过程的行驶路径。

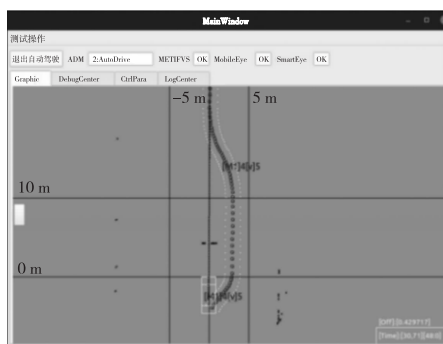


图9 轨迹回放

5 结束语

目前,该避障算法已经在仿真环境中完成多场景的验证测试,并在物流车与观光车等园区专用车辆上进行应用测试,成熟后将在开放道路的载客车辆上应用。

参考文献:

- [1] 彭晓燕,谢浩,黄晶. 无人驾驶汽车局部路径规划算法研究[J]. 汽车工程,2020,42(1):1-10.
- [2] 姚芳,郑帅,吴正斌. 智能车辆局部避障轨迹规划研究[J]. 兵器装备工程学报,2022,43(12):265-271.
- [3] TERAPAPTOMMAKOL W, PHAOHARUHANSA D, KOO-WATTANASUCHAT P, et al. Design of Obstacle Avoidance for Autonomous Vehicle Using Deep Q-Network and CARLA Simulator[J]. World Electric Vehicle Journal, 2018, 13(12):239.
- [4] 乐阳,龚健雅. Dijkstra 最短路径算法的一种高效率实现[J]. 武汉测绘科技大学学报,1999,24(3):209-212.
- [5] 苏卫星,赵晓雯,温永刚,等. 基于环境风险的自动驾驶局部路径规划算法[J/OL]. 信息与控制:1-11[2023-05-04]. <https://doi.org/10.03976/j.cnki.xk.2023.2227>.
- [6] 耿宏飞,神健杰. A* 算法在 AGV 路径规划上的改进与验证[J]. 计算机应用与软件,2022,39(1):282-286.
- [7] 陈成,何玉庆,卜春光,等. 基于四阶贝塞尔曲线的无人车可行轨迹规划[J]. 自动化学报,2015(3):486-496.
- [8] 马玉娇. 基于立体视觉里程计的移动机器人定位研究[D]. 武汉:武汉科技大学,2009.
- [9] WEN Hao, YANG Jiamin. Research on a Costmap that can Change the Turning Path of Mobile Robot[J]. Journal of Physics Conference Series,2005(1):012095.
- [10] ROS Bridge Documentation[EB/OL]. (2022-07-22)[2023-01-10]. <https://carla.readthedocs.io/projects/ros-bridge/en/latest/>.
- [11] 崔靛,朱学军. 基于 ROS 的服务机器人智能导航系统研究与仿真[J]. 传感器与微系统,2020,39(2):22-25.