

基于改进混合 A* 算法的自动泊车路径规划方法研究

白俊卿, 魏雪涛, 张红猛

(西安石油大学 计算机学院, 西安 710065)

摘要: 为了解决在自动驾驶过程中短距离自动泊车场景下, 受环境复杂性影响导致传统的 A* 算法和 RS 曲线加速搜索算法难以应用的问题, 提出了一种加入反向搜索算法的改进混合 A* 算法; 利用地图栅格法和 A* 算法计算启发值, 通过检测车身轮廓线与简化后的障碍物线是否相交判断二者能否相撞, 以节省搜索时间; 通过控制 RS 曲线的扩展方向数量, 保证路径的平滑性; 经 MATLAB 仿真垂直入库和侧方泊车场景, 对改进算法与传统算法进行了对比分析, 验证了同等条件下改进的混合 A* 算法在两种仿真场景的平均搜索时间上分别减少 8.18% 和 20.53%, 且能产生更短、更平滑的路径, 从而验证了所提基于反向搜索算法的混合 A* 算法的优越性。

关键词: 自动泊车; 路径规划; 反向搜索; 混合 A* 算法; 障碍物距离代价

Research on Automatic Parking Path Planning Method Based on Improved Hybrid A* Algorithm

BAI Junqing, WEI Xuetao, ZHANG Hongmeng

(School of Computer Science, Xi'an Shiyou University, Xi'an 710065, China)

Abstract: In order to solve the problem that traditional A* algorithm and RS curve accelerated search algorithm are difficult to apply in short-distance automatic parking scene during autonomous driving due to the complexity of environments, an improved hybrid A* algorithm with reverse search algorithm is proposed. The heuristic value is calculated using the map grid method and A* algorithm. By detecting whether the body contour line intersects with the simplified obstacle line, it is determined whether the two will collide, so as to save the search time; The extended directions of the RS curve is employed to ensure the smoothness of the path; Through MATLAB simulation in vertical parking and side parking scenes, the improved algorithm is compared with the traditional algorithm. The results show that under the same conditions, the improved hybrid A* algorithm reduces the average search time in two simulation scenes by 8.18% and 20.53%, respectively, with a shorter and smoother path, thus verifying the superiority of the proposed hybrid A* algorithm based on the reverse search algorithm.

Keywords: automatic parking; path planning; reverse search; hybrid A* algorithm; distance cost of obstacle

0 引言

汽车的使用在人们的日常生活中越来越普遍, 它为我们出行通勤提供了便利, 但也带来了一系列问题, 如车位少、泊车难等。安全泊车要求综合考虑多个因素, 如驾驶员的情绪、技术熟练度以及泊车路径的复杂性

等, 这些因素的不利影响常常导致泊车事故频发。自动泊车系统可以使驾驶员在泊车过程中解放双手, 减少交通事故的发生。在自动泊车系统中, 路径规划系统扮演着至关重要的角色, 一个优越的泊车路径规划算法能够显著提高泊车的效率, 保障泊车的安全性。因此, 优化泊车路径规划算法成为自动泊车技术发展中的关键

收稿日期: 2024-07-15; 修回日期: 2024-08-22。

基金项目: 西安石油大学研究生创新与实践能力的培养计划(YCS23214304)。

作者简介: 白俊卿(1983-), 女, 硕士, 副教授。

通讯作者: 魏雪涛(1999-), 男, 硕士生。

引用格式: 白俊卿, 魏雪涛, 张红猛. 基于改进混合 A* 算法的自动泊车路径规划方法研究[J]. 计算机测量与控制, 2025, 33(1): 226-234.

环节。

路径搜索算法是精准自动泊车路径规划的基础。常见的路径搜索算法包括图搜索算法、基于采样的算法和智能仿生学算法。其中, 图搜索算法^[1]是一种以地图构建为基础进行搜索的方法, 包括栅格地图法和可视图法, 它利用传感器收集环境信息, 将环境划分为网格空间, 将障碍物表示为网格空间中的占用区域, 然后在未被占用的网格中找到连接起点与终点的路径。此外, 该算法在构建地图之后, 通常会与其他算法结合使用。基于采样的算法首先将环境投影到二维平面上, 然后在投影平面上随机选择 N 个采样点, 最后连接这些采样点以形成可能的路径。该算法通过去除与障碍物相交的连线得到可选的路径, 然而生成的路径仅为局部最优解而非全局最优解, 并且由于采样点的随机性, 每一次生成的路径都可能存在差异。智能仿生学算法通过剖析自然界的发展规律进行仿生学研究, 具体包括蚁群算法、神经网络算法、遗传算法等。这些算法通过模仿自然界中生物的行为和进化过程, 从而有效地解决路径规划问题。

现有的路径规划算法仍存在很多不足, 如搜索效率低、搜索时间长、路径不够平滑等。针对最优路径问题, 文献 [2] 首次提出满足汽车运动学的自动泊车混合 A* 路径规划算法, 并通过数值优化对路径进行后处理, 使其规划的路径实现全局最优。但其搜索路径的时间过长, 且搜索效率较低, 故算法本身仍有优化的空间。为了处理规划的路径包含许多不必要的转折点, 文献 [3] 等提出了一种结合人工势场法的新型启发式函数, 该函数同时包含距离信息和障碍物信息。该方法在提高执行效率和减少转折点数量方面表现出优异的性能。文献 [4] 提出将混合 A* 搜索算法与可见性图规划算法相融合, 以在代客泊车的混合 (连续-离散) 环境中找到最短的可能非完整路径, 从而有效缩短了路径搜索的时间, 然而该方法未考虑搜索路径的平滑度。文献 [5] 为了解决路径的平滑问题, 通过修剪丢弃贝塞尔曲线上多余的节点以确保平滑度和连续性, 进一步地, 为了使搜索的路径更加平滑, 针对自动泊车系统任务的高实时性、高精度和高轨迹质量要求, 文献 [6] 通过动态加载障碍物计算狭窄区域的通过性并在规划时引入车辆自身体积, 利用邻域扩展和贝塞尔曲线优化方法来提高轨迹质量, 满足泊车任务对高轨迹质量的要求。文献 [7] 提出了一种基于改进 A* 算法与动态窗口方法 (DWA) 融合的路径规划方法, 在传统 A* 算法的基础上设计启发式函数, 减少搜索节点数及规划路径的转弯次数, 改进节点邻域搜索策略并降低碰撞障碍物的概率。文献 [8] 提出一种能在非结构化环境中生成精确平滑路径的算法, 该方法基于图搜索算法和离散运

动车辆模型, 通过制定局部较短路径改善路径规划结果的平滑度。但其在实际应用时, 未考虑规划路径距离障碍物太近的问题, 因此存在一定的安全隐患。在此基础上, 文献 [9] 提出了一种基于改进 A* 算法的局部运动规划算法, 这种改进算法通过设置冗余安全空间来避免轮廓碰撞, 并在启发式函数设计中考虑了路径弯曲的代价, 提高了路径的平滑度。然而该方法未优化搜索的路径长度, 导致搜索效率较低。基于此, 提出一种加入反向搜索算法的混合 A* 算法, 减少了在垂直入库和侧方泊车两种经典泊车场景下路径搜索的时间和路径长度等, 同时提高了搜索路径的平滑度和算法运行效率。

1 车辆运动学分析

1.1 车辆外轮廓简化

由于车辆在自动泊车过程中, 会在比较狭窄的泊车点做低速大转角运动, 其泊车路径规划易受到车辆外轮廓的影响。因此, 将复杂的车辆外形简化为如图 1 所示的二维矩形, 以便于判断车辆与障碍物的碰撞是否符合汽车运动学的要求。图 1 中, A、B、C、D 分别为车辆四周的 4 个顶点; L 为车身轴距; W 为车身宽度; L_f 、 L_r 分别表示车辆前悬和后悬长度。

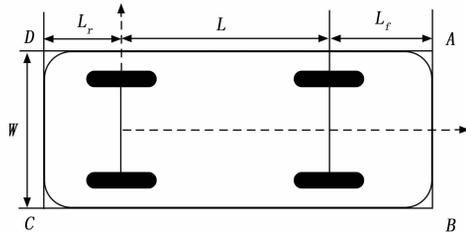


图 1 车辆外形简化图

为了使车辆在泊车过程中尽可能接近实际泊车情况, 参考某国内热门自动驾驶品牌电车, 采用的车辆基本参数如表 1 所示。

表 1 车辆参数表

| 参数 | 符号 | 单位 | 数值 |
|----------|------------------|-----------------------|-------|
| 车辆长度 | L_v | m | 4.500 |
| 轴距 | L | m | 3.700 |
| 车辆宽度 | W | m | 2.600 |
| 车辆前悬 | L_f | m | 0.935 |
| 车辆后悬 | L_r | m | 1.000 |
| 最小转向圆半径 | R_{\min} | m | 5.408 |
| 最大等效前轮转角 | φ_{\max} | rad | 0.600 |
| 最大前轴转角转速 | ω_{\max} | rad · s ⁻¹ | 0.600 |

1.2 车辆运动学建模

为了精准计算车辆转弯时各个车轮的转向角度, 实现良好的转向性能和最小转弯半径。基于二自由度自行车模型和阿克曼转向模型^[10], 建立了如图 2 所示的车

辆运动学模型。

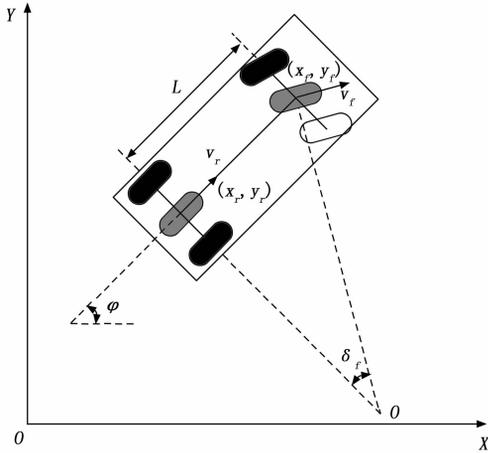


图 2 车辆运动学模型

图 2 中, (x_f, y_f) 、 (x_r, y_r) 分别表示车辆前后轴中心点的坐标。对于低速自动泊车车辆而言, 在前后轮都是纯滚动的理想状态下, 车辆后轴的运动学约束为:

$$\bar{x}_r \sin\varphi - \bar{y}_r \cos\varphi = 0 \quad (1)$$

式中: φ 为二维坐标系下车体的航向角, \bar{x}_r 、 \bar{y}_r 为对时间求导的后轴中心点坐标。

由图 2 所示的几何关系可得车辆前后轴中心点的坐标关系为:

$$\begin{cases} x_r = x_f - L \cos\varphi \\ y_r = y_f - L \sin\varphi \end{cases} \quad (2)$$

对式 (2) 关于时间求导:

$$\begin{cases} \bar{x}_r = \bar{x}_f + L\dot{\varphi} \sin\varphi \\ \bar{y}_r = \bar{y}_f - L\dot{\varphi} \cos\varphi \end{cases} \quad (3)$$

式中: \bar{x}_f 、 \bar{y}_f 表示对时间求导的前轴中心点速度, $\dot{\varphi}$ 表示车辆航向角对时间求导的角速度。

将式 (3) 代入式 (1) 中, 得:

$$\bar{x}_f \sin\varphi - \bar{y}_f \cos\varphi + L\dot{\varphi} = 0 \quad (4)$$

由图 2 的几何关系可知, 前轴中心点速度为:

$$\begin{cases} \bar{x}_f = v_f \cos(\varphi + \delta_f) \\ \bar{y}_f = v_f \sin(\varphi + \delta_f) \end{cases} \quad (5)$$

式中: v_f 为车辆前轴中心速度; δ_f 为等效前轮转角, 向左为正, 向右为负, 故图 2 中 δ_f 为负值。

进一步地, 将式 (5) 代入式 (4) 中, 得:

$$\dot{\varphi} = \frac{v_f}{L} \sin\delta_f \quad (6)$$

将式 (5) 和式 (6) 代入式 (3) 中, 计算得:

$$\begin{cases} \bar{x}_r = v_f \cos\varphi \sin\delta_f \\ \bar{y}_r = v_f \sin\varphi \cos\delta_f \end{cases} \quad (7)$$

由于车辆在运动时, 同一时刻前后轴纵向车速相同, 因此车辆前轴、后轴中心速度 v_f 与 v_r 满足:

$$v_r = v_f \cos\delta_f \quad (8)$$

将式 (8) 代入式 (6) 和式 (7), 可得关于车辆后轴中心的行驶运动学方程为:

$$\begin{bmatrix} \bar{x}_r \\ \bar{y}_r \\ \dot{\varphi} \end{bmatrix} = \begin{bmatrix} \cos\varphi \\ \sin\varphi \\ \frac{\tan\delta_f}{L} \end{bmatrix} v_r \quad (9)$$

由式 (9) 可知, 当车速 v_r 不变, 车辆后轴中心点的位姿仅与车辆轴距 L 和等效前轮转角 δ_f 相关。

车辆在泊车行驶过程中, 由于要考虑车位边界与障碍物的碰撞情况, 因此还需计算车辆外轮廓简化的 4 个顶点 A、B、C、D 的坐标。

图 3 所示为车身定点图, 图中, θ 为车身航向角。由于在泊车过程中, 车辆始终保持低速状态, 因此可忽略质心侧偏角, 认为车辆横摆角 φ 与车身航向角 θ 相等。故可计算出车辆外轮廓 4 个顶点的坐标, 其中 A 点坐标可表示为:

$$\begin{cases} x_A = x_f + (L + L_f) \cos\theta - \frac{W}{2} \sin\theta \\ y_A = y_f + (L + L_f) \sin\theta + \frac{W}{2} \cos\theta \end{cases} \quad (10)$$

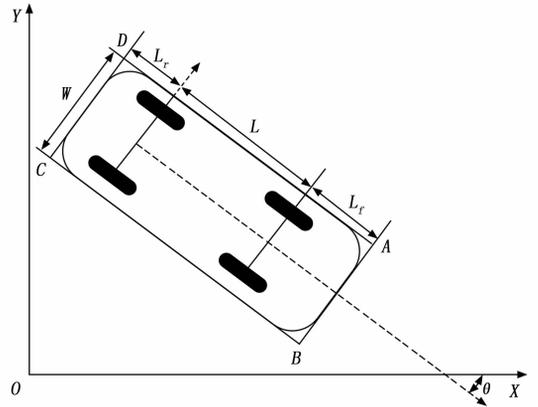


图 3 车身定点图

2 自动泊车路径规划算法

2.1 碰撞检测

在自动泊车系统中, 包围盒^[11]被广泛用于实现碰撞检测以提高计算效率。考虑到文章采用混合 A* 算法进行路径规划, 且道路路况相对简单, 因此采用包围盒与栅格地图法相结合可以更好地描绘地图信息, 即车辆将感知到的起始点坐标与障碍物等转化为二维栅格地图, 再利用得到的地图进行路径规划。

包围盒通常是指车辆周围的虚拟边界框区域, 系统利用这些信息来确保车辆在泊车操作中的安全。这些包围盒覆盖了车辆的前、后、左、右和角落等位置, 其通常是一个围绕物体的最小矩形或立方体, 以便于简化碰撞检测过程。常用的包围盒包括轴对齐包围盒 (AABB,

axis-aligned bounding box)、定向包围盒 (OBB, oriented bounding box) 和球包围盒, 3 种包围盒的具体形式如图 4 所示。其中, AABB 包围盒是一种简单、高效与坐标轴对齐的包围盒类型。OBB 包围盒是有向包围盒, 其比 AABB 包围盒更加接近于物体的真实形状, 占据面积小, 但是计算复杂, 不太适用于动态物体。球包围盒是用一个三维球体完全包裹物体, 通常用于近似物体的形状, 尤其适用于对称的物体, 但它的紧密性太差, 包裹后仍会留下较大的空隙, 需要花费大量的时间进行预处理。

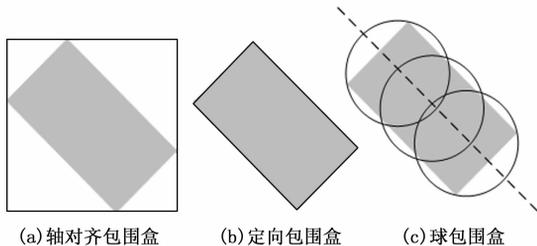


图 4 3 种常见的包围盒

3 种包围盒的对比形式如图 5 所示, 通过对比发现, AABB 包围盒更易计算且更高效。已知车身 4 个顶点的栅格坐标后, 直接利用栅格坐标就可以覆盖此包围盒, 在车辆运行时, 通过旋转 (根据车辆的方向) 和平移 (根据车辆的位置) 来与障碍物进行碰撞检测, 因此更适用于文章设置的垂直入库和侧方泊车两种经典泊车场景。

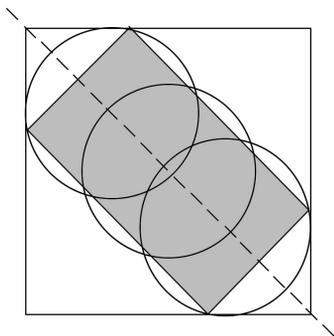


图 5 3 种包围盒对比图

由于研究的是在垂直入库和侧方泊车两种经典泊车场景下, 车辆的短距离路径搜索, 因此假设在车辆搜索和泊车过程中不会遇到过于复杂的障碍物, 重点在于研究泊车过程中的避障操作。研究中将停车点周围的所有边框都设为障碍物, 同时将停车点的入口打开, 具体示意图 6。首先, 根据车辆的位置和方向计算出车辆的轮廓矩形, 接着利用分离轴定理^[12] (SAT, separating axis theorem) 进行碰撞检测。具体而言, 首先将车辆的轮廓矩形边界投影到障碍物所在方向上, 然后通过检

测投影后的区间是否有重叠来判断车辆投影和障碍物是否发生碰撞, 再进一步检测车辆轮廓矩形的边界与障碍物之间是否会碰撞。最终如果任何一条边界与障碍物线相交, 则判断发生碰撞。

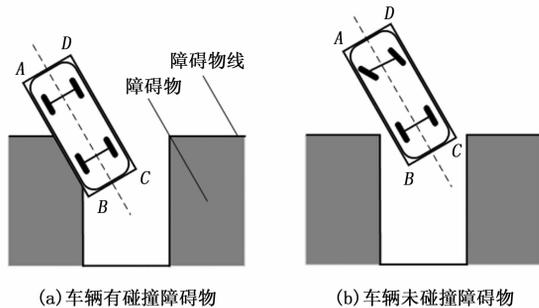


图 6 障碍物线碰撞检测简化图

这种基于分离轴定理的碰撞检测方法, 通过计算车辆轮廓矩形并利用 SAT 算法检测其与障碍物之间的相交情况, 简化了车辆碰撞检测过程, 为车辆在垂直入库和侧方泊车两种经典泊车场景下提供了安全保障且提高了泊车效率。

由于在实际泊车环境中, 不仅存在静态障碍物, 还存在动态障碍物的影响, 故在车辆规划出泊车路径后, 需对动态障碍物的影响进行进一步的判断。当车辆按照规划路径进行泊车行驶过程中, 若动态障碍物运动到与泊车路径相交后, 即车辆检测到所规划路径中有障碍物时, 则停止等待; 当动态障碍物离开规划路径后, 即与所规划的路径不再相交时, 车辆继续运行, 直至运动至终点位置。所设规避动态障碍物的策略, 进一步提高了本文所提路径规划算法的实用性, 提升了车辆在真实环境中的避障和决策能力, 为实现安全自动驾驶提供了基础保障。

2.2 基于反向搜索的混合 A* 算法

A* 算法被归类为确定性智能寻径算法^[13], 采用的贪心策略使其在路径搜索问题中表现尤为出色, 相较于广度优先搜索和深度优先搜索算法而言, A* 算法具有更高的搜索效率。

A* 算法在自动泊车系统应用时, 其对环境建模的要求极高, 故算法将实际环境转换为可以使用并解析的二维栅格地图, 此方法对路径规划提供尤为重要, 栅格地图法在操作上更为简单并容易实现, 被广泛应用于描述自动汽车环境地图中。然而, 传统的 A* 算法采用折线搜索极不符合车辆运动学约束条件^[14]。由于混合 A* 算法具备前进/后退两种模式, 因此采用如图 7 所示的具有 5 个方向的节点扩展方式, 即在汽车保持最小转弯半径的条件下, 对转角进行平均分割, 并检查当前车辆位姿到子节点车辆位姿的变换是否会与障碍物发生碰撞。在此过程中, 利用 Open_Set 表和 Close_Set 表对

节点进行剪枝，同时借助启发式代价函数来选择最优的扩展节点，从而搜索出最佳路径。传统 A* 算法与混合 A* 算法节点拓展方式的对比如图 8 所示。

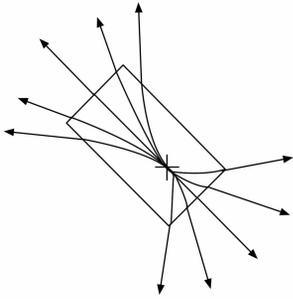


图 7 节点拓展方式图

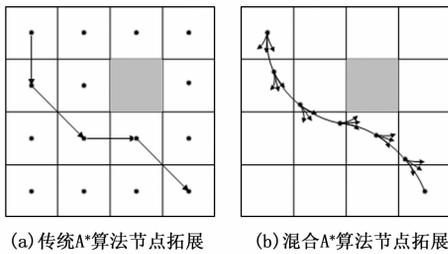


图 8 传统 A* 算法与混合 A* 算法节点拓展对比图

A* 算法的代价函数为：

$$f(n) = g(n) + h(n) \quad (11)$$

式中， $f(n)$ 表示节点 n 的总代价值； $g(n)$ 表示从起始节点到节点 n 的实际代价； $h(n)$ 表示从节点 n 到目标节点的启发值，一般用于表示从节点 n 到目标节点的估计距离的启发函数。混合 A* 算法结合了 A* 算法的启发式搜索和其他算法的特点，以提高搜索效率和路径质量。混合 A* 算法的代价函数 $f(n)$ 会调整 $g(n)$ 和 $h(n)$ 的权重^[15-16]，或者引入其他因素。不同权重之间占比不同，车辆逼近目标达到的效果不同：若增大 $h(n)$ 的权重值，可以使搜索更倾向于使用启发式估计，从而更快地逼近目标。若注重路径平滑性或其他约束，通过调整 $h(n)$ 的权重值，可以在路径规划中更注重特定的约束或优化目标，如路径平滑性或避免障碍物。

常用的启发函数包括欧几里得距离、曼哈顿距离和切比雪夫距离等^[17]。其中，欧几里得距离能够有效优化搜索路径，并兼顾全局信息，其数学性质也较为优良。因此，选择欧几里得距离作为启发函数，采用的欧几里得距离公式为：

$$Distance = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \quad (12)$$

式中， (x_1, y_1) 为当前点坐标， (x_2, y_2) 为终点坐标。

通常情况下，混合 A* 算法从起始位姿开始朝着目标位姿扩展节点，在不碰撞障碍物的前提下，采用 RS 曲线^[18]来连接当前节点和目标节点，以加快搜索速度。

然而，在环境复杂的停车点或者车位之间距离较窄的情况下，RS 曲线可能因为需要避障而过多地扩展节点，从而导致算法搜索时间过长，整体效率不高。为了提高混合 A* 算法在搜索路径时的时效性，采用了一种更为实用的方法，即基于反向搜索的混合 A* 算法，如图 9 所示。这种反向搜索算法是一种图搜索算法，分为两段：第一段（E-O 段）从终点向起始点探索，保证不与障碍物接触，第二段（O-S 段）从当前节点向起点探索，并通过 RS 曲线进行连接。最终，将所有节点逆序输出，生成的完整路径即为反向搜索算法的最佳路径。通过这种方式可以有效提高路径搜索效率，从而减少搜索的时间。

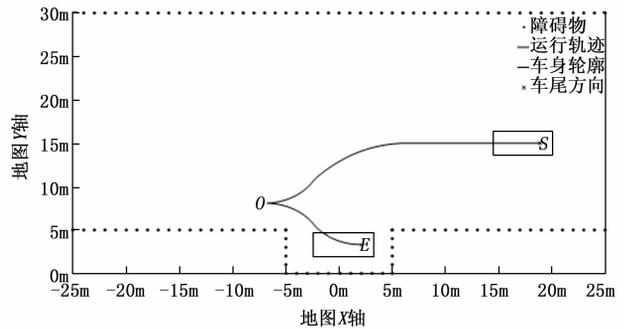


图 9 反向路径搜索图

2.3 路径搜索算法框架

文章提出的加入反向搜索算法的混合 A* 算法，其正向搜索过程的流程图如图 10 所示。具体步骤如下。

步骤 1：首先，设定车辆、运动分辨率和地图形状等参数信息，确定起始位姿和终点位姿的坐标信息，并对栅格地图进行初始化。

步骤 2：初始化表 Open_Set1、Open_Set2、Close_Set1 和 Close_Set2，分别用于存放车辆运动中的拓展节点、已选取父节点和起点及终点方向最小代价子节点。

步骤 3：计算出拓展节点的最小代价值并存入 Close_Set1 中，在混合 A* 算法的基础上利用栅格^[19]索引出下一节点并计算出代价值。

步骤 4：判断 Close_Set2 中是否含有当前拓展节点？若否，则跳转至步骤 6；反之，进入下一步。

步骤 5：比较当前节点和栅格内 Close_Set2 中节点的代价值是否小于等于当前节点和 Open_Set2 中节点的代价值？若是，则删除 Close_set2 中有当前栅格索引的节点，并合并 Close_Set1 和 Close_Set2 后输出最终路径；反之，则继续计算下一拓展节点的最小代价值。

步骤 6：判断当前节点是否在 Open_Set2 中？若否，则将下一节点的父节点设置为当前节点，并更新 Open_Set1 中下一节点的代价值；反之，合并 Close_

Set1 与 Close_Set2, 输出最终路径。

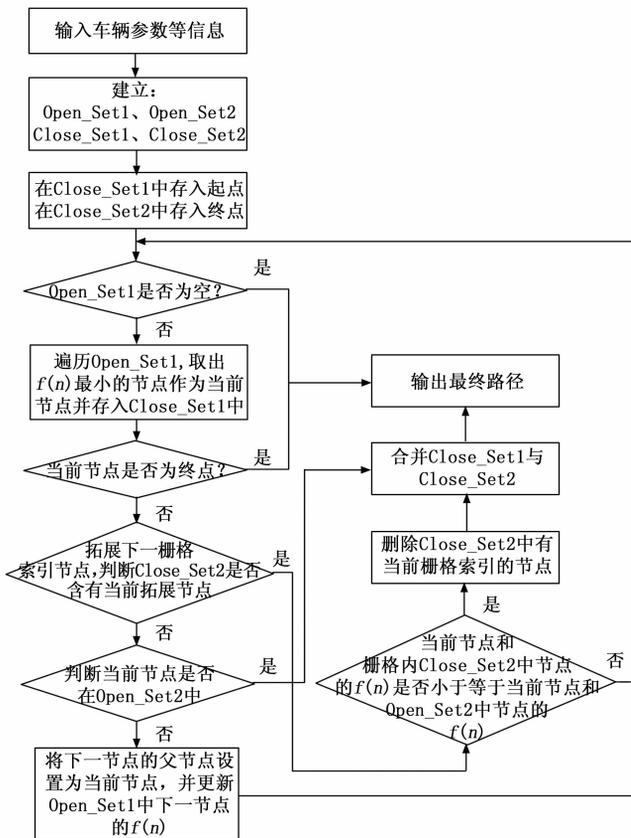


图 10 改进算法正向搜索流程图

3 自动泊车模拟仿真分析

文章使用 MATLAB R2023a 构建仿真平台, 在代码文件中建立 25m × 25m 仿真场景, 创建车辆、障碍物等对象的初始状态, 包括车辆位置、车身和车尾方向并设置车辆的初始速度为匀速 2m/s; 创建车辆的运动学模型并依次设置参数, 考虑车辆动力学约束和优化启发式函数, 编写改进的混合 A* 算法的核心逻辑, 包括节点扩展、启发式函数计算和路径回溯; 最后运行仿真, 并进行数据采集和记录。在仿真设置中, 分别模拟垂直入库场景和侧方泊车场景下路线搜索与真实车辆泊车过程, 并选取符合动力学逻辑的代价最小路径, 通过仿真对比传统算法和改进算法的搜索时长、车辆改变方向次数、节点数和路径长度等, 验证改进的混合 A* 算法相较于传统算法在垂直入库和侧方泊车两种经典泊车场景中的优越性和时效性。

3.1 垂直入库实验与分析

垂直入库实验利用 MATLAB 仿真平台模拟了自动泊车中的垂直入库过程^[20]。该实验主要关注路径规划的长度、车辆转向曲率以及前进/后退的行为, 而不考虑车辆速度和受力^[21]等, 得到传统混合 A* 算法的路径曲线图如图 11 所示。

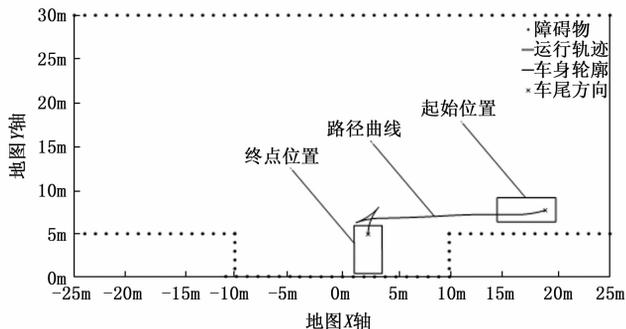


图 11 传统混合 A* 算法路径曲线图

在垂直停车过程中, 起始位姿为 $(22, 8.19, \pi)$, 终点位姿为 $(2.5, 5, -\pi/2)$ 。通过图 11 的模拟可知, 由于存在转角倒车的情况, 传统混合 A* 算法生成的路径图并非最优。其虽然符合车辆运动学要求, 但在这种情况下实际代价较高, 因此得到的最终代价值不一定是最小代价值, 生成的路径也不一定是最短路径。在更换改进算法后, 保持车辆环境不变, 重复相同的实验步骤。通过实验操作, 得到改进算法生成的路径曲线图, 图 12 和图 13 分别是在无障碍物和有障碍物时生成的路径曲线图。

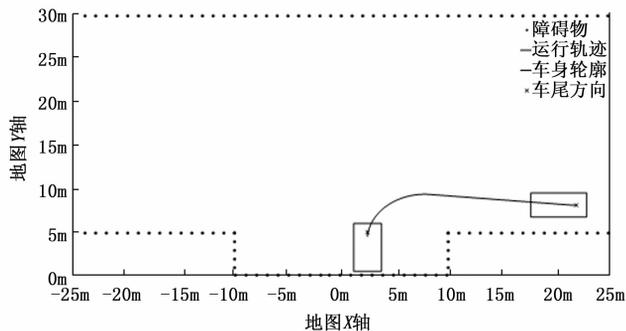


图 12 改进算法路径曲线图

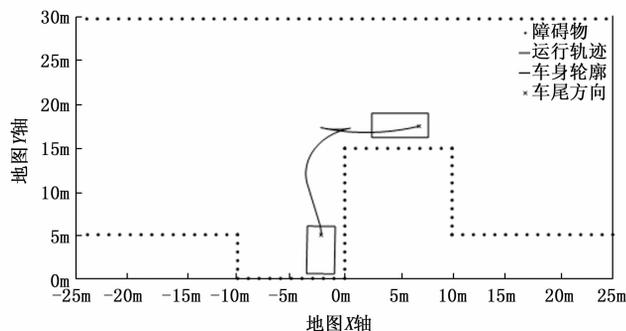


图 13 改变障碍物路径曲线图

通过分析图 11 与图 12, 可得出在相同条件下, 与传统混合 A* 算法相比, 改进的混合 A* 算法生成的路径曲线极大减少了路径改变次数、运行轨迹相对更加平滑、且无停车和倒车的现象。其不仅降低了改进算法中

实际路径总代价值，而且优化了路径搜索时间。进一步对比图 12 和图 13 有无障碍物的两种结果可以发现，改进算法生成的路径受到障碍物的影响较大。随着地图上障碍物复杂程度的增加，路径规划的时间往往会有所波动。这种情况得到的结果不利于直观地比较数据之间的差异。通过保持相同起点位姿为 $(22, 8.19, \pi)$ ，终点位姿为 $(2.5, 5, -\pi/2)$ 后，设置运动分辨率为 0.1，车辆默认速度为 2 m/s，比较传统混合 A* 算法和改进混合 A* 算法的搜索时间、改变方向次数、节点数和路径长度，其结果见表 2。

表 2 垂直入库算法对比

| | 时间/s | 改变方向次数/次 | 节点数/个 | 路径长度/m |
|------|-------|----------|-------|--------|
| 传统算法 | 12.05 | 2 | 242 | 24.1 |
| 改进算法 | 11.35 | 1 | 228 | 22.7 |

根据表 2、图 12 与图 13 的结果可以得出，在相同条件下，改进的混合 A* 算法相较于传统混合 A* 算法，在垂直入库场景下的搜索时间分别减少了 5.81%，并且搜索过程中所访问的节点数也相对较少，生成的路径更加平滑且更短。

为了降低本文实验在不同方向路径上垂直入库实验的偶然性，故在图 11 地图中，选择不同方向的几组起点和终点进行验证，其结果见表 3~6。

表 3 垂直入库算法泊车时间对比

| 起点位姿 | 终点位姿 | 泊车时间/s | |
|--------------------|--------------------|------------|------------|
| | | 经典混合 A* 算法 | 改进混合 A* 算法 |
| $(24, 8.4, \pi)$ | $(2.5, 5, -\pi/2)$ | 14.113.2 | |
| $(-20, 14.5, \pi)$ | $(-7.5, -\pi/2)$ | 10.6 | 9.55 |
| $(16, 18, \pi)$ | $(-2.5, -\pi/2)$ | 12.75 | 11.6 |
| $(-16, 7.5, \pi)$ | $(7, 5, -\pi/2)$ | 14.2 | 13.15 |

表 4 垂直入库算法改变方向次数对比

| 起点位姿 | 终点位姿 | 改变方向次数/次 | |
|--------------------|--------------------|------------|------------|
| | | 经典混合 A* 算法 | 改进混合 A* 算法 |
| $(24, 8.4, \pi)$ | $(2.5, 5, -\pi/2)$ | 2 | 1 |
| $(-20, 14.5, \pi)$ | $(-7.5, -\pi/2)$ | 2 | 1 |
| $(16, 18, \pi)$ | $(-2.5, -\pi/2)$ | 2 | 0 |
| $(-16, 7.5, \pi)$ | $(7, 5, -\pi/2)$ | 2 | 1 |

表 5 垂直入库算法节点数对比

| 起点位姿 | 终点位姿 | 节点数/个 | |
|--------------------|--------------------|------------|------------|
| | | 经典混合 A* 算法 | 改进混合 A* 算法 |
| $(24, 8.4, \pi)$ | $(2.5, 5, -\pi/2)$ | 263 | 247 |
| $(-20, 14.5, \pi)$ | $(-7.5, -\pi/2)$ | 213 | 192 |
| $(16, 18, \pi)$ | $(-2.5, -\pi/2)$ | 256 | 233 |
| $(-16, 7.5, \pi)$ | $(7, 5, -\pi/2)$ | 285 | 274 |

表 6 垂直入库算法路径长度对比

| 起点位姿 | 终点位姿 | 路径长度/m | |
|--------------------|--------------------|------------|------------|
| | | 经典混合 A* 算法 | 改进混合 A* 算法 |
| $(24, 8.4, \pi)$ | $(2.5, 5, -\pi/2)$ | 26.2 | 24.6 |
| $(-20, 14.5, \pi)$ | $(-7.5, -\pi/2)$ | 21.2 | 19.1 |
| $(16, 18, \pi)$ | $(-2.5, -\pi/2)$ | 25.5 | 23.2 |
| $(-16, 7.5, \pi)$ | $(7, 5, -\pi/2)$ | 28.4 | 27.3 |

由表 3~6 可以得出，在相同条件下，改进的混合 A* 算法相较于传统混合 A* 算法，在垂直入库场景下的平均搜索时间减少 8.18%。

3.2 侧方泊车实验与分析

侧方泊车是指在道路旁停车，与垂直入库实验过程类似，文章通过对改进混合 A* 算法和传统混合 A* 算法进行路径规划得出路径曲线，如图 14 与图 15 所示。

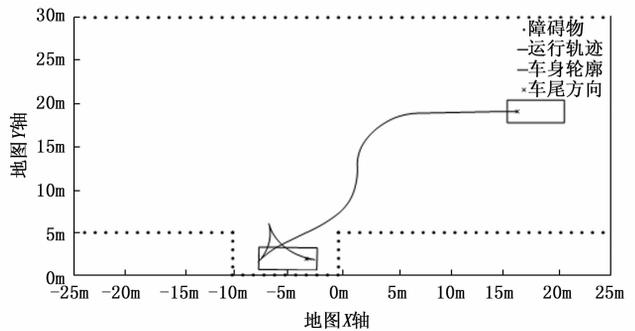


图 14 传统混合 A* 算法搜索路径图

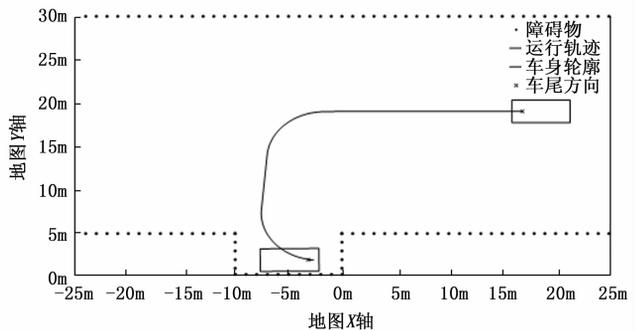


图 15 改进混合 A* 算法搜索路径图

在进行侧方泊车的过程中，起点位姿为 $(20, 19.06, \pi)$ ，终点位姿为 $(-3, 2, \pi)$ 。由图 14 可知，在使用传统混合 A* 算法生成路径时，也存在汽车方向变化导致路径改变的现象，在车位中重复路径过多，路径总消耗代价值太大，无法达到最优路径选择。与图 15 对比发现，在使用改进算法后，车辆在规划路径过程中有效减少了总代价值且路径更平滑、无过多重复，更接近于最优路径的选择标准。

进一步地，通过改变起点位姿，比较传统混合 A*

算法和改进算法在侧方停车情况下的搜索时间、改变方向次数、节点数和路径长度变化, 结果见表 7。

表 7 侧方入库算法对比

| | 时间/s | 改变方向次数/次 | 节点数/个 | 路径长度/m |
|------|------|----------|-------|--------|
| 传统算法 | 23 | 3 | 461 | 46 |
| 改进算法 | 20.9 | 1 | 419 | 41.8 |

由表 5、图 14 与图 15 可知, 在设置相同起始点的侧方泊车场景下, 改进混合 A* 算法会比传统混合 A* 算法的搜索时间减少 9.13%, 且在方向改变次数和搜索路径长度方面明显减少。

同理, 为了降低本文实验在不同方向路径上侧方泊车实验的偶然性, 故仍在图 11 地图上, 选择不同方向的几组起点和终点进行验证, 其结果见表 8~11。

表 8 侧方入库算法泊车时间对比

| 起点位姿 | 终点位姿 | 泊车时间/s | |
|--------------------|-----------------|------------|------------|
| | | 经典混合 A* 算法 | 改进混合 A* 算法 |
| (18, 15.7, π) | (-3, 2, π) | 18.95 | 14.15 |
| (16, 14.7, π) | (2, 2, π) | 14.7 | 13.05 |
| (19, 15.5, π) | (-4, 3, π) | 13.35 | 10.1 |
| (-8, 14.2, π) | (7, 3, π) | 15.65 | 12.5 |

表 9 侧方入库算法改变方向次数对比

| 起点位姿 | 终点位姿 | 改变方向次数/次 | |
|--------------------|-----------------|------------|------------|
| | | 经典混合 A* 算法 | 改进混合 A* 算法 |
| (18, 15.7, π) | (-3, 2, π) | 4 | 2 |
| (16, 14.7, π) | (2, 2, π) | 3 | 1 |
| (19, 15.5, π) | (-4, 3, π) | 3 | 1 |
| (-8, 14.2, π) | (7, 3, π) | 4 | 1 |

表 10 侧方入库算法节点数对比

| 起点位姿 | 终点位姿 | 节点数/个 | |
|--------------------|-----------------|------------|------------|
| | | 经典混合 A* 算法 | 改进混合 A* 算法 |
| (18, 15.7, π) | (-3, 2, π) | 380 | 284 |
| (16, 14.7, π) | (2, 2, π) | 295 | 262 |
| (19, 15.5, π) | (-4, 3, π) | 268 | 203 |
| (-8, 14.2, π) | (7, 3, π) | 314 | 251 |

表 11 侧方入库算法路径长度对比

| 起点位姿 | 终点位姿 | 路径长度/m | |
|--------------------|-----------------|------------|------------|
| | | 经典混合 A* 算法 | 改进混合 A* 算法 |
| (18, 15.7, π) | (-3, 2, π) | 37.9 | 28.3 |
| (16, 14.7, π) | (2, 2, π) | 29.4 | 26.1 |
| (19, 15.5, π) | (-4, 3, π) | 26.7 | 20.2 |
| (-8, 14.2, π) | (7, 3, π) | 31.3 | 25 |

由表 8~11 可以得出, 在相同条件下, 改进的混合 A* 算法相较于传统混合 A* 算法, 在侧方泊车场景下的平均搜索时间减少 20.53%。

综合垂直入库和侧方泊车两种实验场景的仿真结果, 可以得到: 结合反向搜索的混合 A* 算法会比传统的混合 A* 算法在相同场景下搜索时间更少, 且搜索的路径更短、更平滑, 充分证实了混合 A* 算法在结合反向搜索算法后, 自动泊车路径规划的效率得到显著提高。

3.3 动态障碍物规避检测实验与分析

动态障碍物避障是指车辆泊车环境在同时拥有静态障碍物与动态障碍物的前提下, 进行的路径规划策略, 如图 16 所示。

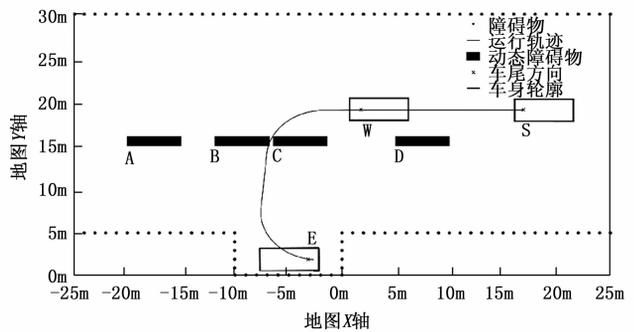


图 16 传统混合 A* 算法搜索路径图

在进行动态障碍物规避检测的过程中, 起点位姿为 (20, 19.06, π), 终点位姿为 (-3, 2, π), 由图 14 可知, 在地图 Y 轴为 15 m, 地图 X 轴为 (-20 m, 5 m) 的区间中增加一个长 5 m, 宽 1 m 的动态障碍物, 并使此障碍物从 A 点以速度为 2 m/s 的在此区间作水平单向运动, 直至 D 点结束; 在此期间, 车辆从起点位姿 S 点运动到终点位姿 E 点, 动态障碍物运行到 B 点时, 与车辆规划路径相交, 车辆停止在 W 点处等待动态障碍物离开, 直至其运动到 C 点, 车辆则继续运行, 最终动态障碍物在 D 点停止, 车辆在 E 点泊车。

在车辆路径规划研究中, 动态障碍物的引入能够更真实地模拟实际道路环境中的复杂性和不确定性。通过在仿真中加入动态障碍物, 车辆需要在复杂多变的环境中进行决策, 这可以有效考察和验证路径规划算法的优越性。

4 结束语

1) 提出一种基于反向搜索算法的混合 A* 算法, 其与地图栅格化相结合, 能够显著提高自动泊车路径规划的效率。改进算法相比传统算法在两种泊车场景中分别平均减少了 8.18% 和 20.53% 的搜索时间, 并能产生更短、更平滑的路径。

2) 实验结果表明, 在现实生活中的垂直入库和侧方泊车场景下, 采用基于反向搜索的混合 A* 自动泊车算法能够降低节点数、车辆改变方向次数和路径长度, 从而更有效地解决了路径规划问题, 为自动泊车系统的实际应用提供了更好的解决方案。

一个全面的自动泊车实用路径规划系统应包括路径规划、对地图空车位的感知、最优车位的定位、动态障碍物最优避障策略以及车辆的速度变速控制, 由于只对自动泊车路径规划算法进行研究, 未对地图空车位感知系统和泊车过程的空车位感知进行分析, 这也是下一步该研究的内容。

参考文献:

- [1] NIEMEYER K E, SUNG C J. On the importance of graph search algorithms for DRGEP-based mechanism reduction methods [J]. *Combustion and Flame*, 2011, 158 (8): 1439 - 1443.
- [2] DOLGOV D, THRUN S, MONTEMERLO M, et al. Practical search techniques in path planning for autonomous driving [J]. *Ann Arbor*, 2008, 1001 (48105): 18 - 80.
- [3] LI X, LI G, BIAN Z. Research on Autonomous Vehicle Path Planning algorithm Based on Improved RRT* algorithm and Artificial Potential Field Method. [J]. *Sensors (Basel, Switzerland)*, 2024, 24 (12): 3899 - 3899.
- [4] SEDIGHI S, NGUYEN D V, KUHNERT K D. Guided hybrid A-star path planning algorithm for valet parking applications [C] //2019 5th international conference on control, automation and robotics (ICCAR). *IEEE*, 2019: 570 - 575.
- [5] ELHOSENY M, THARWAT A, HASSANIEN A E. Bezier curve based path planning in a dynamic field using modified genetic algorithm [J]. *Journal of Computational Science*, 2018, 25: 339 - 350.
- [6] ZHAO Y. Automatic parking planning control method based on improved A* algorithm [J]. *ArXiv Preprint ArXiv: 2406.15429*, 2024.
- [7] LI Y, JIN R, XU X, et al. A mobile robot path planning algorithm based on improved A* algorithm and dynamic window approach [J]. *IEEE Access*, 2022, 10: 57736 - 57747.
- [8] CHU K, KIM J, JO K, et al. Real-time path planning of autonomous vehicles for unstructured road navigation [J]. *International Journal of Automotive Technology*, 2015, 16: 653 - 668.
- [9] MIN H, XIONG X, WANG P, et al. Autonomous driving path planning algorithm based on improved A* algorithm in unstructured environment [J]. *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, 2021, 235 (2 - 3): 513 - 526.
- [10] 王秋, 曲婷, 陈虹. 基于随机模型预测控制的自主车辆转向控制 [J]. *信息与控制*, 2015, 44 (4): 499 - 506.
- [11] 李星, 傅妍芳, 王亮, 等. 基于射线检测的动态碰撞优化算法 [J]. *系统仿真学报*, 2019, 31 (11): 2393 - 2401.
- [12] TAZAKI Y, OKUDA H, SUZUKI T. Parking trajectory planning using multiresolution state roadmaps [J]. *IEEE Transactions on Intelligent Vehicles*, 2017, 2 (4): 298 - 307.
- [13] WU Z, DAI J, JIANG B, et al. Robot path planning based on artificial potential field with deterministic annealing [J]. *ISA Transactions*, 2023, 138: 74 - 87.
- [14] PAPPAS G J, KYRIAKOPOULOS K J. Stabilization of non-holonomic vehicles under kinematic constraints [J]. *International Journal of Control*, 1995, 61 (4): 933 - 947.
- [15] 赵锦涛, 李亮, 薛仲瑾, 等. 基于混合 A 星的停车场内巡航分层运动规划方法 [J]. *机械工程学报*, 2023, 59 (24): 290 - 298.
- [16] Abdel-Rahman A S, Zahran S, Elnaghi B E, et al. Enhanced hybrid path planning algorithm based on apf and a-star [J]. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 2023, 48: 867 - 873.
- [17] 蔡佳成, 白克强, 李旭春, 等. 基于 JPS 改进的机器人路径规划算法 [J]. *计算机应用研究*, 2022, 39 (7): 1985 - 1991.
- [18] KIM M, AHN J, PARK J. Targettree-rrt*: Continuous-curvature path planning algorithm for autonomous parking in complex environments [J]. *IEEE Transactions on Automation Science and Engineering*, 2022, 21 (1): 606 - 617.
- [19] DU Z, MIAO Q, ZONG C. Trajectory planning for automated parking systems using deep reinforcement learning [J]. *International Journal of Automotive Technology*, 2020, 21 (4): 881 - 887.
- [20] Vorobieva H, Glaser S, Minoiu-Enache N, et al. Automatic parallel parking in tiny spots: Path planning and control [J]. *IEEE Transactions on Intelligent Transportation Systems*, 2014, 16 (1): 396 - 410.
- [21] SONG P, XIE S, ZHONG Y, et al. A unified creep-speed control approach for automated parking system [R]. *SAE Technical Paper*, 2017.