

基于行车风险场的变邻域A*算法车辆路径规划

胡林治, 叶 洲, 黄陈雨佳

上海理工大学光电信息与计算机工程学院, 上海

收稿日期: 2024年4月15日; 录用日期: 2024年5月15日; 发布日期: 2024年5月22日

摘 要

随着车辆智能化水平的飞速发展, 为了满足智能车辆对全局路径规划的速度、平滑性和安全性要求, 提出了一种基于行车风险势场的变邻域改进A*算法。首先对栅格地图的环境建模中加入必要安全空间和冗余安全空间, 保留足够的行车安全裕度; 为了提升环境适应性, 基于车辆当前位置和目标点的位置关系, 改进了启发函数的计算方式; 同时结合行车风险势场对于车辆产生的合力, 改进算法扩展搜索机制, 引入了横纵向和斜向增扩策略, 提升算法的搜索效率; 完整路径规划算法完成后, 通过冗余点剔除和动态切点光滑算法对路径进行二次处理, 形成最终的平滑车辆路径。最后通过仿真实验进行算法对比, 验证了改进算法对于智能车辆路径规划性能和行车安全性的提升。

关键词

智能车辆, 路径规划, A*算法, 启发函数, 行车风险势场

Vehicle Path Planning Using Variable Neighborhood A* Algorithm Based on Driving Risk Field

Linzhi Hu, Zhou Ye, Chenyujia Huang

School of Optoelectronic Information and Computer Engineering, University of Shanghai for Science and Technology, Shanghai

Received: Apr. 15th, 2024; accepted: May. 15th, 2024; published: May. 22nd, 2024

Abstract

With the rapid development of vehicle intelligence, a variable neighborhood improved A* algorithm based on the driving risk field is proposed to meet the speed, smoothness, and safety re-

requirements of intelligent vehicles for global path planning. Firstly, necessary safety spaces and redundant safety spaces are added to the environmental modeling of raster maps, while retaining sufficient driving safety margins; In order to improve environmental adaptability, the calculation method of the heuristic function was improved based on the relationship between the current position of the vehicle and the position of the target point; At the same time, combining the combined force of driving risk potential field on vehicles, the algorithm extension search mechanism is improved, and horizontal, vertical, and diagonal expansion strategies are introduced to enhance the search efficiency of the algorithm; After the completion of the complete path planning algorithm, the redundant points are removed and the dynamic tangent smoothing algorithm is used to perform secondary processing on the path, forming the final smooth vehicle path. Finally, algorithm comparison was conducted through simulation experiments to verify the improvement of the algorithm in intelligent vehicle path planning performance and driving safety.

Keywords

Intelligent Vehicles, Path Planning, A* Algorithm, Heuristic Function, Driving Risk Potential Field

Copyright © 2024 by author(s) and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

1. 引言

随着我国汽车工业的崛起,无论是从国家层面还是社会企业层面都加大了对智能车辆的研发和投入,智能化技术也在不断蓬勃发展。智能车辆是集环境感知、决策规划以及辅助驾驶等功能于一体的高度自治的系统,融合了传感器、计算机、通信、人工智能还有自动控制等多种技术[1]。车辆路径规划作为集中体现车辆智能化水平的重要模块功能,一直是行业内研究的焦点。

全局路径规划是指在由高精地图、车联网提供了已知行驶环境情况下,在全局地图上规划出一条从车辆起始位置到目标位置的点对点的高效、安全且舒适的行驶路径,使车辆能够跟随规划路线,实现行程目标。具体来说即在包含了静态障碍物和可行域信息的地图上[2],搜索出一条从初始点出发到目标点通行效率最高的路线。全局路径规划主要包含两方面内容,一是环境的建模,二是路径点搜索算法。合适的环境建模,地图构建是进行路径规划的基础,要去除环境中的干扰因素,只保留会对车辆行驶造成影响的环境边界和障碍物信息,在提升规划质量的同时降低对计算资源的占用。本文假设全局环境信息已由高精地图获取,在已有环境建模的基础上对进行更进一步的处理。

常用的路径规划算法包括基于栅格图的搜索规划,如 Dijkstra 算法、D*算法、A*算法等,基于采样的规划有快速随机搜索树和概率路线图,还包括基于智能算法的规划,如粒子群算法、遗传算法、蚁群算法等。目前已有很多研究对 A*算法进行了改进。文献[3]提出的任意时间树恢复加权 A*算法是对加权 A*算法的改进,该算法以牺牲部分最优性为代价加速了搜索过程,并通过重用以前的搜索信息减少了搜索时间。文献[4]提出了一种斜八邻域扩展策略,在原八邻域扩展策略的基础上,在 x 轴和 y 轴方向上多扩展一个节点,再将其和四邻域扩展方式结合,融合成一种新的双邻域选择扩展策略。文献[5]中使用了加权 A*算法,通过增加了可调节的权重系数,增强了启发式函数的影响,从而提高了算法执行效率,但结果容易陷入局部最优。文献[6]针对 A*算法中的启发式函数难以确定或不一致,导致算法性能不稳定的问题,提出了多启发式 A*算法,通过不同启发式函数的组合来改变启发式函数的引导能力,从而简化了启发式函数的设计,提高了算法的性能,避免了局部极小值的出现。

本文的研究目标就是在不同行驶场景下根据确定的出发点，以到终点的行驶安全性和距离代价最小为目标进行全局规划。在参阅了相关文献后，发现现有研究对于 A* 算法多以目标点为导向进行应用和改进，而没有考虑障碍物对于算法搜索的潜在影响，因此加入行车风险势场对算法的节点扩展搜索方向进行引导。针对算法中陷入局部最优空间，搜索策略效率低，转折点多、路径不平滑等问题，对算法代价函数计算和规划结果分别进行优化改进。

2. 全局规划地图构建

2.1. 栅格地图建立

进行全局路径规划首先要确定环境信息，建立环境地图，根据得到的障碍物和可行域在方位上的布局，构建合理的障碍物分布图，然后在此基础上通过规划搜索算法进行路径寻优，合理的环境地图可以更好地帮助确定规划方法和搜索算法，最终以更少的资源代价规划出更优的路径。初步的环境建模首先对获取到的卫星地图进行预处理，将行驶环境映射成一个整合了环境特点和障碍物轮廓的地理表示。

常用的环境建模方法包括栅格法、几何法和拓扑法。文献[7]使用拓扑图来描述环境，增加了规划路径和障碍物之间的安全距离，但仍然存在拓扑信息来源困难以及需要额外碰撞检测的问题。还有研究[8]以圆形边界来包围车身轮廓以形成一个安全距离约束，只有满足了安全约束的路径点才会被使用，但这样的边界判定增加了计算量。栅格图中每个栅格都对应着真实工作环境中的具体位置，栅格大小都是确定的相同的，易于数据存储检索和处理。

获取到的卫星地图不能直接应用于路径搜索算法，首先要将环境模型转化为栅格图，忽略所有物体的高度信息，只保留长宽信息，过程主要是将全域环境以一定的精度/分辨率划分为大小相等的正方形区域，每一个方形区域就是栅格地图的基本计算单位，栅格的尺寸越小，栅格地图对环境的描述越精确。同时给每一块方形区域都赋予不同的属性。

栅格地图的数学模型[9]可以表示如式(1)：

$$A = \sum_{i=1}^m \sum_{j=1}^n a_{ij}, \quad a_{ij} = 0, 1 \quad (1)$$

式中：A 为整个环境地图的模型，m 和 n 分别为环境地图栅格化后每行、每列的栅格数， a_{ij} 为栅格地图中 (i, j) 处的栅格位置， $a_{ij} = 0$ 表示栅格处可通行， $a_{ij} = 1$ 表示栅格处有障碍物，无法通行。

2.2. 增加安全空间的环境建模

由于在栅格地图中使用 A* 算法进行路径搜索时使用的是车辆点质量运动模型。而汽车不同于移动机器人，机器人体积相对较小，栅格的大小通常和机器人的轮廓体积相近，可保证机器人不和障碍物发生碰撞。但是车辆规划中如果采用和机器人同样的处理方法，必然会造成规划和实际的偏差。忽略车身体积，实际无法通过的路径可能会在规划中被视为可通行。不考虑转弯半径，规划中的拐点可能会造成实车的激进转向。因此需要在环境模型中补偿因忽略车身体积而造成的误差。

同时，实际行驶过程中还存在陷阱区域，即障碍物形成了一个内凹的可行驶区域，如果算法搜索的路径点进入区域后，算法可能会在陷阱区域内规划出激烈的转弯甚至掉头路径，又或者在较大的陷阱区域内行驶一段路程后再次回到起点。对安全空间进行处理，具体就是对障碍物的边界进行膨胀扩展，保留一定的安全行车空间。

对栅格地图进行安全空间处理易于计算，且不需要降低地图分辨率。本文将安全空间的优化分为必要安全空间和冗余安全空间两部分，前者是保证车辆和障碍物之间安全无碰撞的必要距离，同时为局部轨迹规划留有一定的空间裕度。后者是为了防止路径搜索算法在陷入陷阱区域后导致规划路径扭曲变形，

造成车辆无法跟踪的情况发生。必要安全空间的设置步骤如图 1(a)所示,在规划中采用车辆点质量模型,但实际中要考虑车身体积,图中黄色方框代表车辆。为了降低陷阱区域的影响,对障碍物再进行冗余安全空间设置。在栅格地图中填补陷阱区域,设置其为潜在障碍物。冗余安全空间的设置,让一些不能保证车辆安全通过的狭窄通道和陷阱空间被冗余空间覆盖,提升了规划路线的整体平滑性。具体安全空间设置示意如图 1(b)所示。

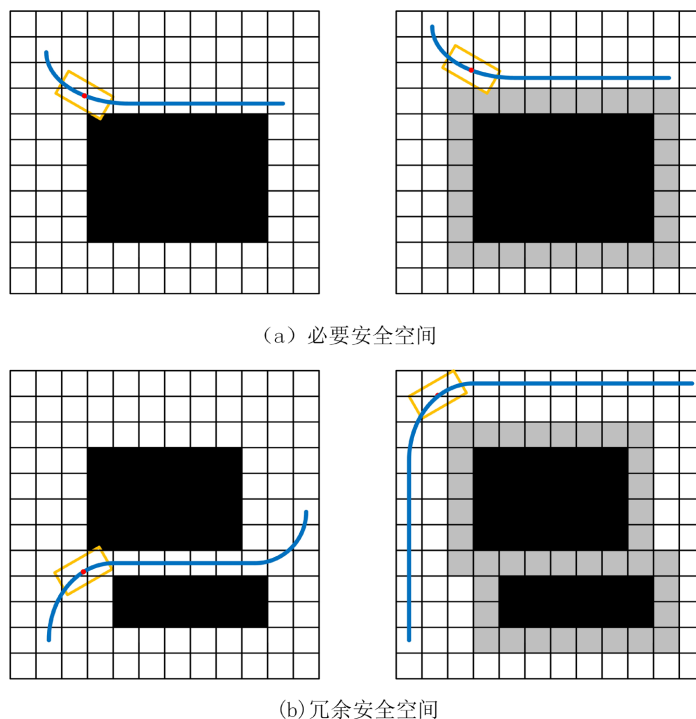


Figure 1. Schematic diagram of safety space setting
图 1. 安全空间设置示意图

3. A*算法和行车风险场

目前 A*算法在移动机器人和车辆规划中已经有了很多应用,本文在改进 A*算法的同时,引入了障碍物对车辆产生的风险势场影响。

3.1. 传统 A*算法

A*算法是在 Dijkstra 算法的基础上扩展而来的启发式算法[10],在代价计算中,相比于 Dijkstra 算法, A*算法在计算实际代价的基础上,加入了启发式函数,从而定义了新的代价函数,使得 A*算法在完成二维空间中的路径搜索和寻优时可以充分发挥效力。A*具体计算方式如式(2):

$$f(n) = g(n) + h(n) \quad (2)$$

式中 $g(n)$ 表示从起始点到当前节点的实际行驶路径代价, $h(n)$ 表示从当前节点到目标点的预估路径代价,也就是启发函数; $f(n)$ 表示当前节点总的预估代价,用以评判当前节点是否代价最优。

首先要明确的就是启发函数的计算方式,通常的距离计算方式有四种:曼哈顿距离、欧几里得距离、切比雪夫距离和对角线距离[11]。对角线距离就是可以沿着栅格对角线移动的距离加上不能沿着对角线移动的距离,允许在八邻域方向上移动。如果四邻域搜索的代价为 D ,那么对角线代价 $D_2 = \sqrt{2}D$,具体对

角线距离计算如式(3):

$$\begin{cases} h_1(n) = \min[abs(x_n - x_g), abs(y_n - y_g)] \\ h_2(n) = abs(x_n - x_g) + abs(y_n - y_g) \\ h(n) = D_2 * h_1(n) + D * [h_2(n) - 2 * h_1(n)] \end{cases} \quad (3)$$

确定了启发函数就可以进行 A*算法的具体实施, 算法流程如图 2 所示:

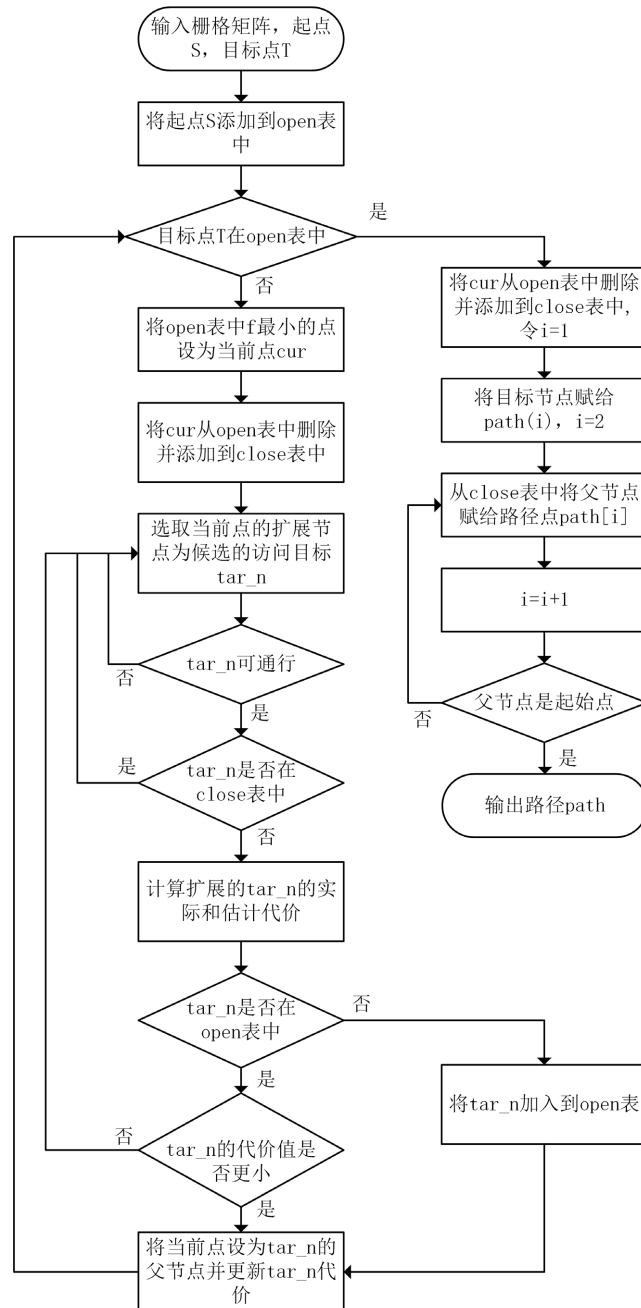


Figure 2. A* Algorithm flowchart
图 2. A*算法流程图

3.2. 行车风险势场建立

车辆行驶过程中除了目标方位决定了行驶方向，同时还面临着障碍物、障碍车对行车方向的扰动影响，需要根据车辆定位和所处环境进行行驶风险的评估，提升行车安全性。合理的行车势场可以将障碍对于车辆的影响数值化，形成具体的势场力。参考文献[12]，类比物理电场建立行车势场，势场力的计算方式参考人工势场法[13]，具体如下。

目标点势场如式(4)：

$$U_{att} = \frac{1}{2} k_1 \rho^2(p, p_g) \quad (4)$$

其中 k_1 为正比例增益系数， $\rho(p, p_g)$ 为一个矢量，表示汽车的位置 p 和目标点位置 p_g 之间的距离矢量，大小为 $|p - p_g|$ ，方向是从汽车的位置指向目标点的位置。

相应的引力 F_{att} 为目标势场场的负梯度如式(5)：

$$F_{att} = -\nabla U_{att} = k_1 \rho(p, p_g) \quad (5)$$

决定障碍物风险势场的因素是汽车和障碍物之间的距离，当车辆不在障碍物的影响范围内，其受到的势能值为 0，在汽车进入障碍物的影响范围后，两者之间的距离越大，汽车受到的势能就越小，距离越小，汽车受到的势能值就越大。风险势场的势场函数如式(6)：

$$U_{req} = \begin{cases} \frac{1}{2} k_2 \left(\frac{1}{\rho(p, p_0)} - \frac{1}{\rho_0} \right)^2 & 0 \leq \rho(p, p_0) \leq \rho_0 \\ 0 & \rho(p, p_0) \geq \rho_0 \end{cases} \quad (6)$$

其中 k_2 为正比例系数， ρ_0 为一常数，表示障碍物对汽车产生影响的距离。

相应的斥力 F_{rep} 如式(7)：

$$F_{rep} = k_2 \left(\frac{1}{\rho(p, p_0)} - \frac{1}{\rho_0} \right) \frac{1}{\rho^2(p, p_0)} \quad (7)$$

用行车势场对车辆形成的合力来预测行车方向，改进路径点的搜索机制，可以充分考虑到障碍物的影响，保证充足的避让空间的同时，又确保了前进的方向性。

4. 基于行车风险场的改进 A*算法

A*算法进行路径寻优时搜索速度快，但同时也会有陷入局部最优的问题。并且随着环境复杂程度的增加，问题越凸显，算法的执行效率将受到很大的减益。

4.1. A*算法改进

4.1.1. 启发函数改进

启发函数决定着算法的搜索行为，如果只使用四方位的曼哈顿距离，那么会增加规划路径中的转折，降低路径效率，如果只采用全方位的欧式距离，那么规划出的路径可能会出现较大的拐点，这又不符合车辆运动特性。本文采用对角线距离作为启发函数的计算公式，并针对直线和对角线距离加上不同的优化处理，具体计算方式如下：

$$\begin{cases} d_1 = \max(|x_n - x_g|, |y_n - y_g|) \\ d_2 = \min(|x_n - x_g|, |y_n - y_g|) \end{cases} \quad (8)$$

通过上式(8)判断出当前位置和目标点位置的横向和纵向距离差。

$$\begin{cases} h_1(n) = \sqrt{2}d_2 \\ h_2(n) = d_1 + d_2 \\ h_3(n) = d_1 - d_2 \end{cases} \quad (9)$$

式(9)中 $h_1(n)$ 和 $h_2(n)$ 分别对应对角线距离中的计算公式, $h_3(n)$ 用于判断车辆在当前位置的运动倾向。当 $h_1(n) \geq h_3(n)$ 时, 整个行程更偏向于斜向运动, 启发函数计算如式(10):

$$h(n) = w_1 * \sqrt{2} * h_1(n) + w_2 * [h_2(n) - 2h_1(n)] \quad (10)$$

当 $h_1(n) < h_3(n)$ 时, 表明整个行程更偏向于直线方向运动, 启发函数计算如式(11):

$$h(n) = w_2 * \sqrt{2} * h_1(n) + w_1 * [h_2(n) - 2h_1(n)] \quad (11)$$

通过调整启发函数中的权重值来改变 A* 算法的搜索行为, 当两个权重值都为 0 时, 算法就会退化为 Dijkstra 算法, 规划得到的路径最短, 但所花费的计算代价也最大, 扩展次数多, 计算时间长。随着两个权重值的增大, 算法的扩展次数会呈现指数级下降, 但过大的权重又会导致算法退化为 BFS, 导致路径最优性下降。因此, 考虑到启发函数预估距离应该小于等于当前节点到路径目标点的真是距离, 两个权重值之和不应该过大。改进后的启发函数代价取值为 $w_1 = 2$, $w_2 = 1$ 。

4.1.2. 引入代价权重

确定好启发函数后, 再对整体代价函数的计算方式进行优化。由以上不同情况下的路径搜索结果分析, 以及所参考的文献, 采用变权重代价函数来改进启发函数, 优化后的具体的代价和权重计算公式如下式(12):

$$\begin{cases} f(n) = g(n) + w(n) * h(n) \\ w(n) = 1 + \frac{d(n)}{d(s)} \end{cases} \quad (12)$$

式中, $d(n)$ 表示当前点到目标点距离, $d(s)$ 表示起始点到目标点的距离, 搜索起始的阶段, 权重公式中 $d(n)/d(s)$ 的值趋近于 1, 这时权重系数就相对大, 有利于算法的搜索; 当算法运行到行程最后, 比值就趋近于 0, 实际路程代价和预估路程代价的权重就接近, 更适合算法的寻优。

4.2. 引入势场导向的改进 A* 算法

常用的 A* 算法所采用的节点扩展方法都为 4 邻域或者 8 邻域扩展, 为了进一步增加行车的安全性, 在保证方向性的同时, 还应该保证车辆行驶方向不会靠近障碍物。因此将行车势场和 A* 算法相结合, 提出基于行车风险场的变邻域扩展 A* 路径搜索算法。进一步提升算法的搜索效率, 减少搜索节点个数。改进后的算法在父节点进行搜索时判断势场合力方向, 然后根据合力方向有针对性地搜索栅格。

根据当前点所受合力方向与地图坐标系的斜率 k 来确定算法的节点扩展方向, 同时加入当前位置相对于目标点方位的考虑, 当斜向距离更大, 节点应尽可能斜向扩展, 扩展方向如表 1, 示意如图 3; 当横向距离更大时, 扩展应更偏向直线方向搜索, 扩展方向如表 2, 示意如图 3。

确定了扩展搜索的方向后, 采用变邻域增扩搜索策略进一步对节点扩展方式进行改进, 首先基于对角线距离判定当前车辆所处位置距离目标点是斜向距离大还是直线距离大。若斜向距离大, 那么采用斜向增扩策略, 根据合力方向判定所属区域, 如表 1 所示, 每一个区域都包含三个扩展栅格, 除此以外, 当斜向对角上第一个栅格没有障碍物时, 进行斜向增扩, 即再向外扩展一个对角栅格, 如图 4 中(c)所示,

否则保持原扩展搜索策略。若直线距离大，那么采用横纵向增扩策略，同样先根据合力方向判定所属区域，如表 2 所示，每一个区域也都包含三个扩展栅格，当前栅格的横纵方向上第一个扩展栅格没有障碍物时，进行横纵向增扩，即再向外扩展一个横向或纵向的栅格，具体如图 4 中(d)所示，否则保持原扩展搜索策略。

Table 1. Table of slope and direction correspondence during oblique direction
表 1. 斜向时斜率和方向对应表

k 值	扩展方向	k 值	扩展方向
$(-1/3, 1/3), F_x \geq 0$	1	$(-1/3, 1/3), F_x < 0$	5
$[1/3, 3], F_x \geq 0$	2	$[1/3, 3], F_x < 0$	6
$(3, +\infty) \cup (-\infty, -3), F_y \geq 0$	3	$(3, +\infty) \cup (-\infty, -3), F_y < 0$	7
$(-3, -1/3), F_x < 0$	4	$(-3, -1/3), F_x \geq 0$	8

Table 2. Table of slope and direction correspondence in both horizontal and vertical directions
表 2. 横纵向时斜率和方向对应表

k 值	扩展方向
$[-1, 1], F_x \geq 0$	A
$(1, +\infty) \cup (-\infty, -1), F_y \geq 0$	B
$[-1, 1], F_x < 0$	C
$(1, +\infty) \cup (-\infty, -1), F_y < 0$	D

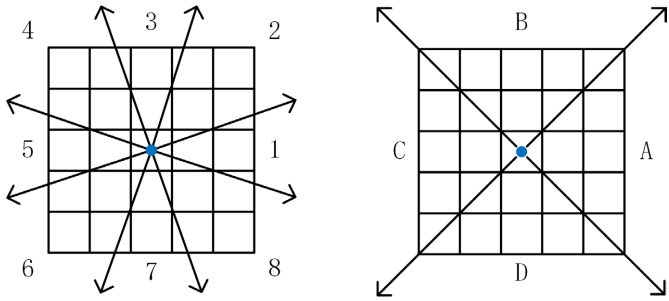


Figure 3. Schematic diagram of expanding search direction
图 3. 扩展搜索方向示意图

5. 规划轨迹二次处理

由于栅格地图的限制，上述改进算法最终得到的规划路径仍是以栅格路径点为基础的折线，这不符合实车的行驶状况。如果直接发送给车辆控制器，会导致路径点地重复计算，浪费计算资源，同时也会加剧车辆运动的不稳定性。因此需要对规划完成的路径进行冗余点剔除和平滑优化的二次处理。

5.1. 冗余点剔除

冗余点在一定程度上保证了规划的连续性和路径的安全性，但同时，由于大量冗余点的存在，也导致了规划出的路径并非为最优路径[14]。去除多余的节点有利于车辆以规划路径为基准平稳地运行，具体

步骤如下:

步骤 1: 将规划完成的路径点坐标依次放入集合 A 中, $\{A_i | i=1, 2, 3, \dots, n\}$ 。

步骤 2: 从起始点开始, 连接 A_1A_2 和 A_2A_3 形成向量并判断 $\angle A_1A_2A_3$, 如果角度为 0, 说明节点在同一直线上, 那么 A_2 就是冗余点。如果角度不为 0, 则连接 A_1A_3 , 同时判断之间是否存在障碍物, 如果没有, 那么 A_2 依然判定为冗余点。

步骤 3: 循环步骤 2, 依次递推集合中的节点, 直到集合中的点全部遍历完成剔除冗余点后的效果如图 5 所示。

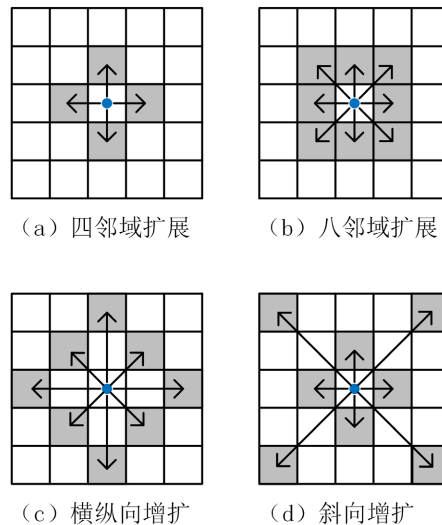


Figure 4. Improve and expand search strategies
图 4. 改进扩展搜索策略

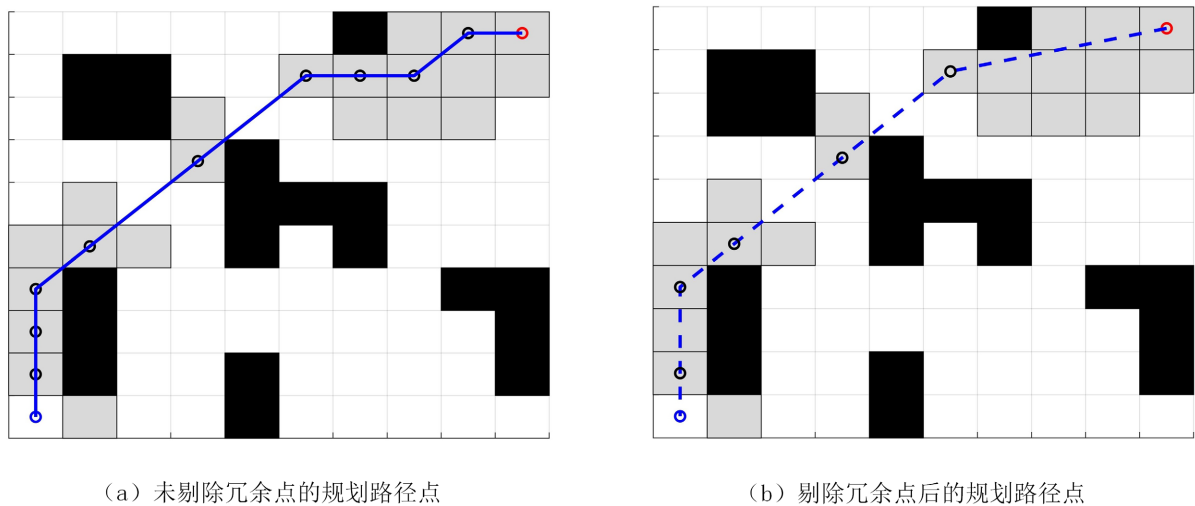


Figure 5. Comparison chart before and after removing redundant points
图 5. 剔除冗余点前后对比图

5.1. 路径平滑

在路径规划中, 当运动方向发生了变化时, 车辆随之也要进行转向动作, 产生一定的转角。在转向

时, 由于方向突然改变, 速度和加速度的方向也会改变。然而在栅格地图的路径规划中, 车辆航向发生变化的转折点处都是折线, 这就导致了转角突然的抖动, 这不符合实际中车辆运动的特性, 不满足车辆动力学约束, 同时剧烈的转向也会降低乘车舒适性[15]。

在路径平滑方面, 现有文献中有两种常见的方法来解决光栅地图的路径平滑问题。第一种是使用特定的曲线进行拟合, 如 Bezier 曲线、B 样条曲线等。该方法通常用于规划完整路径后的路径点拟合。由于曲线拟合不考虑障碍物, 拟合后仍存在碰撞风险, 因此需要在平滑后进行碰撞检测, 对不合格的平滑参数重新调整容易造成重复平滑和碰撞检测。另一种平滑的方法是增加搜索维数或网格搜索范围。例如添加航向角状态, 将状态空间从二维空间变为三维空间。这些方法可以在一定程度上平滑路径, 但容易造成计算量的大量增加。

贝塞尔曲线结构简单、计算效率高[16]。B样条曲线具有贝塞尔曲线的所有优点，同时克服了贝塞尔曲线不能局部修改和连续条件难以获得的缺点[17]。动态切点调整算法通过动态调整路径上的切点来生成平滑且安全的路径，常用于移动机器人和自动驾驶车辆的路径规划，能够生成平滑且安全的路径，适应动态变化的环境，提高路径的实时性和鲁棒性[18]。本文采用动态切点调整算法来去除路径点中的部分不安全或不必要拐点，让路径在满足车辆运动几何特性的同时，保持连续的曲率，具体平滑步骤如下：

步骤 1: 将剔除冗余点的路径点存入集合 B 中, $\{B_i | i=1,2,3,\cdots,n\}$, 从 B_1 开始, 依次遍历集合中的节点。

步骤 2: 比较 $B_{i-1}B_i$ 和 B_iB_{i+1} 线段长度, 以较短线段的未相交端点 S 作为起始点, 作线段的垂线, 和 $\angle B_{i-1}B_iB_{i+1}$ 的平分线相交于 P 。

步骤3: 以相交点为圆心, 端点和交点的连线段为半径作圆。

步骤 4: 判断圆是否和较长线段相交, 如果有交点 Q , 跳转步骤 5, 否则跳转步骤 6。

步骤 5: 判断以圆两个交点为界的圆弧是否经过障碍物, 如果存在障碍物, 跳转步骤 2, 进行下一个点的判定, 如果不存在障碍物, 那么就用圆弧替换折线路径。

步骤 6: 将切点 S 移动到 S_2 , 直到切点圆和较长线段有交点, 再将 S_2 设置为初始点, 并跳转步骤 2, 直到集合中的路径点全部遍历完成。

具体如图 6 所示。

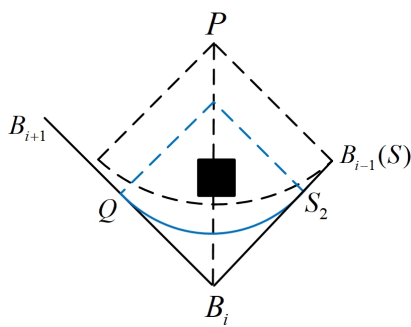


Figure 6. Improve and expand search strategies
图 6. 改进扩展搜索策略

平滑后的路径效果如图7所示。

6. 仿真实验及结果分析

在 MATLAB 中进行仿真实验,并假设已经完成了从实际地图到栅格地图的建模和冗余空间的配置。

为了更好地验证本文改进算法的性能,建立了规则障碍物分布和随即障碍物分布两种仿真地图环境,并分别在 30×30 , 60×60 , 100×100 这三种地图大小下进行仿真验证,起始点设定为(1, 1),终止点分别设定为(30, 30), (60, 60)和(100, 100)。为了保证实验结果的有效性,将本文提出算法和使用曼哈顿距离的四邻域 A*搜索算法,使用欧氏距离的 8 邻域搜索算法进行对比。仿真结果如图 8 和图 9 所示。图中蓝色圆圈表示起始点,红色圆圈代表终止点,黑色圆圈代表剔除冗余节点后的剩余路径节点,蓝色为规划路径,绿色为平滑优化的部分路径。

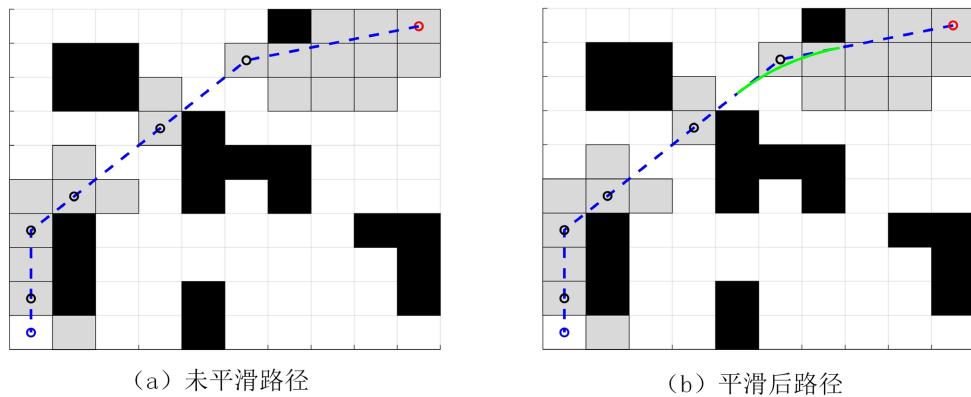


Figure 7. Comparison before and after path smoothing

图 7. 平滑前后路径效果对比

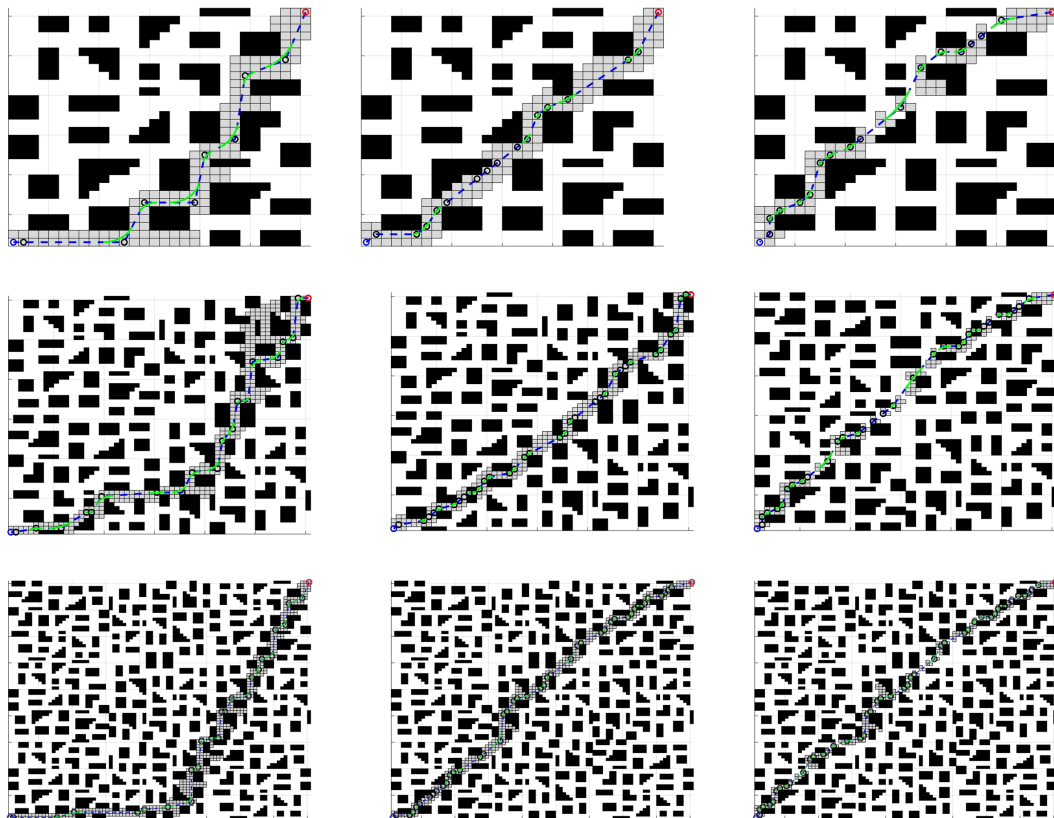


Figure 8. Comparison of planning effects in a rule-based environment

图 8. 规则环境下规划效果对比

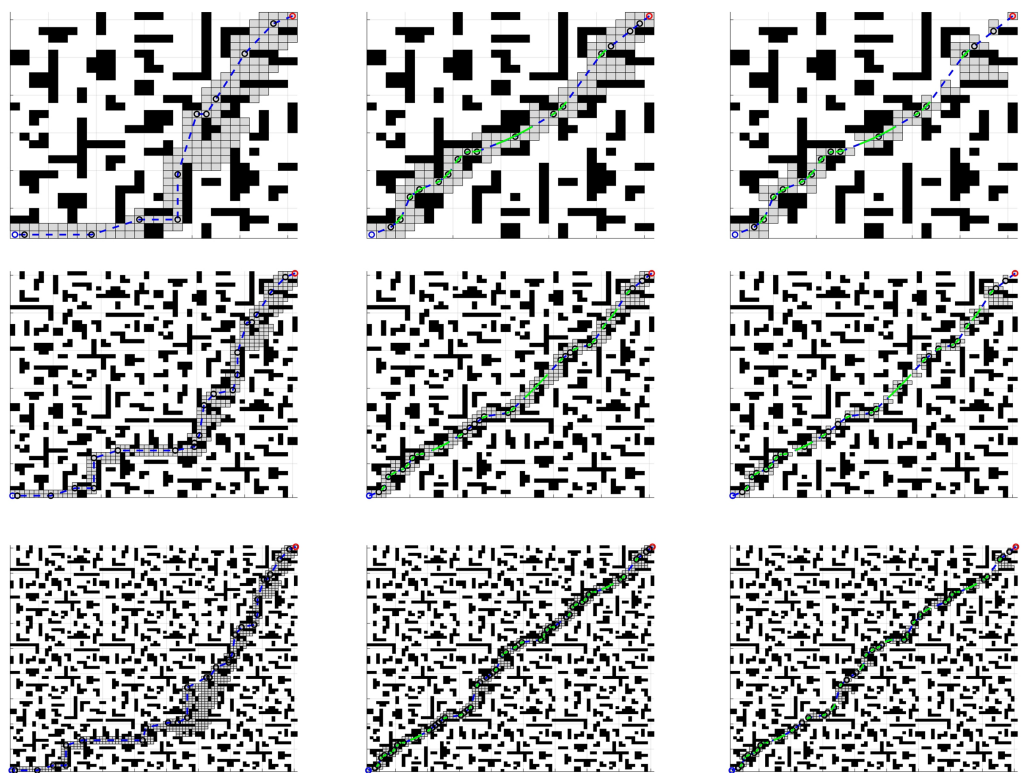


Figure 9. Comparison of planning effects in random environments
图 9. 随机环境下规划效果对比

具体算法对比数据如下。

Table 3.Comparison of planning effect data
表 3.规则环境下规划效果数据对比

数量		四邻域扩展	八邻域扩展	改进 A*
规则环境 (30 × 30)	节点扩展数	164	177	81
	路径长度	49.9324	43.2146	42.2926
规则环境 (60 × 60)	节点扩展数	335	355	155
	路径长度	101.8335	89.2595	88.7633
规则环境 (100 × 100)	节点扩展数	546	599	260
	路径长度	156.4311	135.8619	134.8739

Table 4.Comparison of planning effect data
表 4.随机环境下规划效果数据对比

数量		四邻域扩展	八邻域扩展	改进 A*
随机环境 (30 × 30)	节点扩展数	174	198	83
	路径长度	51.5183	42.5740	42.5740
随机环境 (60 × 60)	节点扩展数	342	391	160
	路径长度	99.1579	87.7071	86.6724

续表

随机环境 (100 × 100)	节点扩展数	688	610	248
	路径长度	140.7975	133.4048	134.1860

从表 3 和表 4 的数据对比可以看出, 本文提出的改进 A*算法节点搜索数大幅降低, 且规划路径的长度也保持在较优的水平。表明本文所提出的改进算法能够在保证路径最优性的同时, 降低算法的计算成本, 提升车辆的路径规划效率。

7. 总结

针对传统 A*算法在车辆全局路径规划应用中存在的不足, 本文首先对环境建模方法进行了优化, 增加了车身和障碍物之间的安全空间。其次, 将周围障碍物对车辆运动方向的影响纳入考虑, 引入行车风险势场来改善和引导 A*算法路径节点的扩展搜索方向, 通过对比实验验证了改进后的针对性路径点搜索在保证节点扩展效率的同时, 路径距离代价也保持在最优水平。最后, 在优化 A*算法的基础上, 对规划路径进行冗余点剔除和路径平滑优化二次处理, 提升路径规划的有效性。仿真实验得出的最终改进规划路线进一步降低了中间冗余节点数量, 平滑后路径节点之间的过渡更加平缓, 并规避障碍物。综上本文所提出的改进算法可以有效地提升 A*算法在智能车辆路径规划中的应用效果。

参考文献

- [1] 吴征, 黄志超, 谷雨阳. 试论我国智能车辆的发展趋势与挑战[J]. 中国设备工程, 2023(z2): 95-97.
- [2] Mac, T.T., Copot, C., Tran, D.T. and De Keyser, R. (2016) Heuristic Approaches in Robot Path Planning: A Survey. *Robotics and Autonomous Systems*, **86**, 13-28. <https://doi.org/10.1016/j.robot.2016.08.001>
- [3] Gochev, K. Safonova, A. and Likhachev, M. (2014) Anytime Tree-Restoring Weighted A* Graph Search. *Seventh Annual Symposium on Combinatorial Search*, Prague, 15-17 August 2014, 80-88.
- [4] 杨秀建, 袁志豪, 白永瑞, 等. 双邻域选择扩展 A*路径规划算法[J/OL]. 机械科学与技术, 1-15. <https://doi.org/10.13433/j.cnki.1003-8728.20230223>, 2024-03-13.
- [5] Min, H., Xiong, X., Wang, P. and Yu, Y., (2021) Autonomous Driving Path Planning Algorithm Based on Improved A* Algorithm in Unstructured Environment. *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, **235**, 513-526. <https://doi.org/10.1177/0954407020959741>
- [6] Aine, S., Swaminathan, S., Narayanan, V., et al. (2016) Multi-Heuristic A*. *The International Journal of Robotics Research*, **35**, 224-243. <https://doi.org/10.1177/0278364915594029>
- [7] Lv, T., Zhao, C. and Bao, J. (2017) A Global Path Planning Algorithm Based on Bidirectional SVGA. *J Robot Journal of Robotics*, **2017**, Article ID: 8796531. <https://doi.org/10.1155/2017/8796531>
- [8] Singh, Y., Sharma, S., Sutton, R., et al. (2018) A Constrained A* Approach towards Optimal Path Planning for an Unmanned Surface Vehicle in a Maritime Environment Containing Dynamic Obstacles and Ocean Currents. *Ocean Engineering*, **169**, 187-201. <https://doi.org/10.1016/j.oceaneng.2018.09.016>
- [9] 熊晓勇. 自动驾驶汽车多场景复杂约束轨迹规划研究[D]: [博士学位论文]. 长春: 吉林大学, 2023.
- [10] Hart, P.E., Nilsson, N.J. and Raphael, B. (1968) A Formal Basis for the Heuristic Determination of Minimum Cost Paths. *IEEE Transactions on Systems Science and Cybernetics*, **4**, 100-107. <https://doi.org/10.1109/TSSC.1968.300136>
- [11] Jing, X. and Yang, X. (2018) Application and Improvement of Heuristic Function in A* Algorithm. 2018 37th Chinese Control Conference (CCC), Wuhan, 25-27 July 2018, 2191-2194. <https://doi.org/10.23919/ChiCC.2018.8482630>
- [12] 闫春来. 自动驾驶车辆局部轨迹规划与跟踪控制算法研究[D]: [硕士学位论文]. 济南: 齐鲁工业大学, 2023.
- [13] Khatib, O. (1985) Real-Time Obstacle Avoidance for Manipulators and Mobile Robots. *Proceedings.1985 IEEE International Conference on Robotics and Automation*, St. Louis, 25-28 March 1985, 500-505. <https://doi.org/10.1109/ROBOT.1985.1087247>
- [14] Chen, H. and Zheng, H. (2022) Research on Full Coverage Path Planning Algorithm of Mobile Robot Based on Astar Improved Algorithm. *Proceedings of the 2022 7th International Conference on Intelligent Information Technology*, Foshan, 25-27 February 2022, 21-27. <https://doi.org/10.1145/3524889.3524893>

-
- [15] Zhang, H., Tao, Y. and Zhu, W. (2023) Global Path Planning of Unmanned Surface Vehicle Based on Improved A-Star Algorithm. *Sensors*, **23**, Article 6647. <https://doi.org/10.3390/s23146647>
 - [16] Long, T. (2023) Improved A-Star Algorithm in Efficiency Enhancing and Path Smoothing. 2023 *IEEE International Conference on Control, Electronics and Computer Technology (ICCECT)*, Jilin, 28-30 April 2023, 28-31. <https://doi.org/10.1109/ICCECT57938.2023.10141188>
 - [17] Li, X., Gao, X., Zhang, W. and Hao, L. (2022) Smooth and Collision-Free Trajectory Generation in Cluttered Environments Using Cubic B-Spline Form. *Mechanism and Machine Theory*, **169**, Article 104606. <https://doi.org/10.1016/j.mechmachtheory.2021.104606>
 - [18] Li, B., Acarman, T., Zhang, Y., Ouyang, Y., Yaman, C., Kong, Q., Zhong, X. and Peng, X. (2021) Optimization-Based Trajectory Planning for Autonomous Parking with Irregularly Placed Obstacles: A Lightweight Iterative Framework. *IEEE Transactions on Intelligent Transportation Systems*, **23**, 11970-11981. <https://doi.org/10.1109/TITS.2021.3109011>