

## 改进 A\* 算法的智能车路径规划研究

杨 瑶<sup>1</sup>, 付克昌<sup>1</sup>, 蒋 涛<sup>1</sup>, 向泽波<sup>2</sup>, 刘甲甲<sup>1</sup>

(1. 成都信息工程大学 控制工程学院, 成都 610225; 2. 中国科学技术大学 软件学院, 合肥 230000)

**摘要:** 由于 A\* 算法所规划的路径存在着转折次数多, 路径不平滑, 路径贴合障碍物和初始时刻转折角度过大等不符合车辆运动学的问题; 为了解决上述问题, 获得适用于智能车的优化路径, 首先通过对车辆运动学进行建模得到车辆的约束, 同时在估价函数中加入车身轮廓代价和障碍物距离代价, 并将车辆约束加入到 A\* 算法的启发函数和路径优化中, 最后使用贝塞尔曲线拟合转折点, 使 A\* 算法所生成的路径更加符合车辆的运动学; 实验结果表明, 改进后的算法所规划的路径更加平滑、合理, 且符合车辆的运动特性。

**关键词:** 智能车; 路径规划; A\* 算法; 车身轮廓代价; 障碍物距离代价

## Research on Intelligent Vehicle Path Planning Based on Improved A\* Algorithm

Yang Yao<sup>1</sup>, Fu Kechang<sup>1</sup>, Jiang Tao<sup>1</sup>, Xiang Zebo<sup>2</sup>, Liu Jiajia<sup>1</sup>

(1. School of Control Engineering, Chengdu University of Information Technology, Chengdu 610225, China;

2. School of Software Engineering, University of Science and Technology of China, Hefei 230000, China)

**Abstract:** There are several problems existing in the path planned by A\* algorithm in real traffic situations. These problems included too many turning points, unsmooth path, path fitting roadside and large turning angle at the beginning, which is not satisfied with the kinematic model of the vehicle. In order to solve the above problems and obtain an optimized path suitable for smart cars, the vehicle constraints are obtained by modeling the vehicle kinematics firstly. Then body contour cost of the car and the obstacle distance cost are added to the evaluation function. At last, the vehicle constraints are added to A\* algorithm's heuristic function and path optimization, and Bezier curves are used to fit the turning points so that the path generated by the A\* algorithm is more suitable for the kinematics of the vehicle. The results of improved A\* algorithm demonstrated that the path planned by the improved algorithm is smoother, more reasonable and suitable for motion characteristics of the vehicle.

**Keywords:** intelligent vehicle; path planning; A\* algorithm; contour cost of intelligent vehicle; distance cost of obstacle

## 0 引言

随着智能驾驶技术的快速发展, 将智能车运用在日常生活中的可能性也越来越大。而智能驾驶技术是实现智能汽车与智能交通的关键技术, 也是未来汽车发展的趋势<sup>[1]</sup>。目前, 学者们对路径规划方法进行了大量研究, 主要包括: 基于采样的快速搜索随机树<sup>[2]</sup>等方法; 基于节点 A\* 算法<sup>[3]</sup>等; 基于模型的人工势场法<sup>[4]</sup>等; 基于生物启发式的蚁群算法等<sup>[5]</sup>。其中, A\* 算法是能够有效求解出最优路径的完备规划算法<sup>[6]</sup>, 也是在机器人上使用最广泛的规划算法之一。但传统 A\* 算法仍存在一些问题。如: 所规划的路径不平滑<sup>[7]</sup>, 路径紧贴障碍物<sup>[8]</sup>和满足车辆运动学等问题。针对上述问题, 本文通过对车辆进行建模分析, 得到车辆

最大转向角约束和最大曲率约束, 并与 A\* 算法的启发函数相结合使得扩展节点的代价值更为合理。同时, 提出车身轮廓代价消除传统 A\* 算法在扩展过程中遍历不合理的节点。并提出障碍物距离代价, 解决了所规划的路径贴合障碍物的问题。然后, 提取出改进 A\* 算法的转折点, 并将车辆最大曲率约束与转折点相结合得到适合智能车行驶的转折点, 再使用贝塞尔曲线拟合转折点得到最优路径。通过上述改进使得 A\* 算法所规划出的路径更加符合车辆的运动学特性并便于实现智能车的路径跟踪。

## 1 车辆的运动学模型

车辆是一个非完整性约束的系统<sup>[9]</sup>, 分析较为困难, 因此本文只考虑车辆的简化运动模型 (如图 2 所示)。本文的实验平台为野马 E70 电动汽车 (如图 1 所示)。在图 2 中的惯性坐标系 XOY 下,  $(x, y)$  为车体在此坐标系下的坐标位置,  $\varphi$  为车体的横摆角 (航向角) ( $\varphi < \varphi_{\max}$ ),  $l = 1.56$  米为智能车的轴距,  $R$  为智能车的转弯半径,  $\theta$  为车辆纵轴与  $x$  轴之间的夹角,  $D = 1.835$  m 为智能车的车宽。本文中只考虑车辆在二维平面上运动, 不考虑俯仰和侧倾带来的影响<sup>[9]</sup>。在车辆运动瞬间, 车辆的速度与车体保持平衡, 即满足:

$$dx * \cos(\theta) - dy * \sin(\theta) = 0 \quad (1)$$

收稿日期:2020-03-01; 修回日期:2020-04-13。

**基金项目:** 四川省科技厅重点研发项目 (2019YFG0188, 2019YJ0413); 四川省科技厅科技计划项目 (2017GZ0069); 成都信息工程大学教改项目 (JY2018021, JY2018053, JY2018118)。

**作者简介:** 杨 瑶 (1994-), 男, 四川绵阳人, 硕士研究生, 主要从事路径规划方向的研究。

蒋 涛 (1975-), 男, 四川绵阳人, 博士, 教授, 主要从事人工智能及智能决策、机器人定位与导航、无人驾驶技术方向的研究。

**通讯作者:** 付克昌 (1970-), 男, 重庆市人, 博士, 副教授, 主要从事路径规划、定位与导航、过程监控方向的研究。

$$K_{\max} = \frac{\tan(\varphi_{\max})}{L} = \frac{1}{R_{\min}} \quad (2)$$



图 1 基于野马 E70 电动汽车改造的智能车驾驶平台

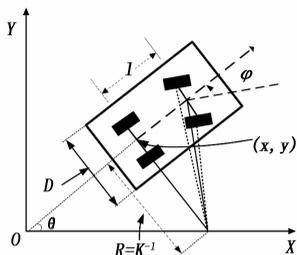


图 2 本文车辆简化运动学模型

$K_{\max} = 16 \text{ m}^{-1}$  车辆最大转弯曲率,  $R_{\min} = 8 \text{ m}$  车辆最小转弯半径,  $\varphi_{\max} = 40^\circ$  车辆最大转向角。

## 2 A\* 算法

A\* 算法是一种在静态路网中求解最优路径的最有效的搜索算法, 也是解决其他搜索问题的有效算法。A\* 算法的评价函数的一般形式为:

$$f(n) = g(n) + h(n) \quad (3)$$

其中:  $f(n)$  是从初始点到节点  $n$  的代价和节点  $n$  到目标点的估计代价的总和,  $g(n)$  是在状态空间中从初始点到节点  $n$  的实际代价,  $h(n)$  是从节点  $n$  到目标点的估计代价。

A\* 算法的基本步骤:

- 1) 在栅格地图上获得起始点和目标点并初始化开启列表和关闭列表。
- 2) 将起始点添加入开启列表中, 并将起始点作为父节点添加到关闭列表中。
- 3) 将父节点四周与障碍物无碰撞的扩展节点添加到开启列表中并计算其扩展节点的代价值, 再判断扩展节点是否为目标点。如果是, 则从关闭列表中得到路径, 结束规划。否则执行步骤 4)。
- 4) 从开启列表中, 取得最小代价值的节点作为父节点并将此节点添加到关闭列表中。再执行步骤 3)。

传统的 A\* 算法的  $h(n)$  多数采用 Manhattan 距离和 Euclidean 距离进行估计, 并未考虑车辆的运动学特性和车辆的体积。因此传统 A\* 所规划出来的路径往往不具有连续的曲率, 且因转折次数过多等导致不满足车辆运动学特性。

## 3 改进 A\* 算法

为了解决传统 A\* 算法所规划的路径不适用于智能车

的问题, 本文对传统 A\* 算法进行改进。首先, 将传统 A\* 算法的启发函数  $h(n)$ , 用连续曲线替代。其次, 提出车轮轮廓代价和障碍物距离代价解决所生成的路径贴合路沿的问题。同时, 对车辆进行运动学建模得到车辆约束条件, 将约束条件带入 A\* 算法所规划路径的转折点中。最后, 使用贝塞尔曲线对转折点进行拟合。

改进 A\* 算法中的评价函数定义为:

$$f(n) = g(n) + h(n) \times (1 + o(n)) + s(n) \quad (4)$$

$$g(n) = \begin{cases} g(n) & (g(n) < o(n)) \\ 0 & (g(n) \geq o(n)) \end{cases} \quad (5)$$

其中:  $o(n)$  是障碍物距离代价,  $s(n)$  是智能车轮廓代价。公式 (5) 中, 当  $g(n) < o(n)$  时,  $g(n)$  取值为起始点到达节点为的代价; 但当  $g(n) \geq o(n)$  时,  $g(n)$  取值为零。通过上述判断是为了让障碍物的安全距离代价能够获得更高的权重。

### 3.1 对函数 $h(n)$ 进行改进

由于传统 A\* 算法的  $h(n)$  一般使用 Euclidean 距离和 Manhattan 距离对扩展节点的代价值进行估计, 所以往往会造成扩展节点被过小估计或者过大估计。比如: Euclidean 距离 (如图 3 (a) 所示) 通过计算扩展节点到目标点的直线距离会造成对  $h(n)$  估计过小的问题。而 Manhattan 距离 (如图 3 (b) 所示) 是由扩展节点和目标点的  $x$  和  $y$  绝对值之差相加所得, 会造成对  $h(n)$  估计过高的问题。因此, 本文使用圆弧曲线对  $h(n)$  进行估计较为合理 (如图 3 (c) 所示)。此外, 由于智能车行驶路线为连续可导的曲线, 而 Euclidean 距离和 Manhattan 距离都是不连续且估计的代价值相较图 3 (c) 所估计的代价值误差较大。

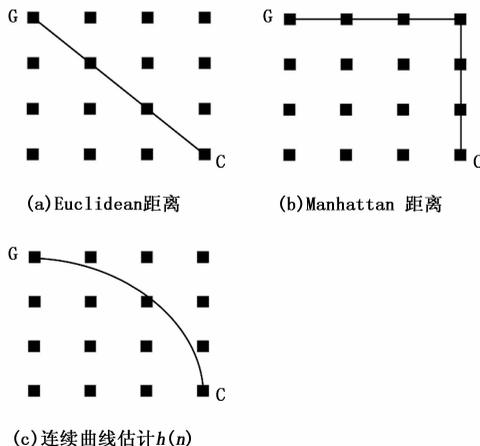


图 3 启发函数  $h(n)$  代价值示意图

为了计算图 3 (c) 中曲线的长度, 本文采用图 4 的方法来进行计算。其中, 曲线  $\widehat{GOC}$  为等腰三角形的弧长, 等腰三角形的底角为  $90^\circ - \varphi_{\max}$ 。同时, 通过图 4 的几何关系可以得到公式 (6) 和公式 (7), 并将  $\widehat{GOC}$  的值赋值给  $h(n)$ 。

$$R = \frac{d}{2 * \cos(90^\circ - \varphi_{\max})} \quad (6)$$

$$\widehat{GOC} = \text{rad}(\angle GOC) * R \quad (7)$$

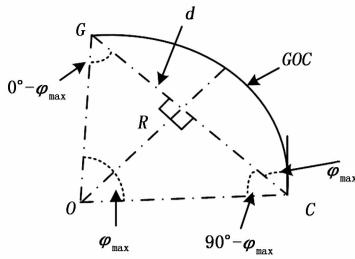


图 4 曲线估计示意图

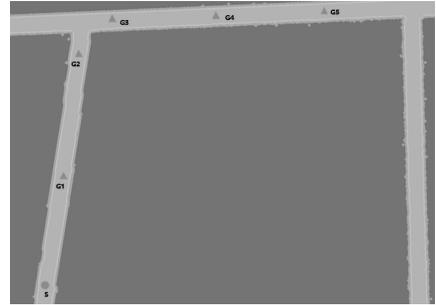


图 7 测试点坐标位置显示

### 3.2 车身轮廓代价 $s(n)$

因为智能车具有一定的体积且存在着最小转弯半径的问题，所以不能当成一个质点处理。同时，传统 A\* 算法在扩展过程中会扩展到贴合障碍物的节点（如图 9 中的扩展节点），这对于智能车来说是没有必要的。此外，传统 A\* 算法规划中未能考虑路径与障碍物的距离。这些问题导致所规划的路径不能很好被智能车所执行。对于这个问题，马飞<sup>[8]</sup>等人提出碰撞威胁代价来解决 A\* 算法所生成的路径贴合障碍物的问题，但是通过在每个扩展节点上加入铲运机 9 个轮廓特征点来计算碰撞代价，会导致程序计算量增加。因此，本文通过判断扩展节点  $p(n) = (x_1, y_1)$  四周距离的障碍物检测点（如图 8 所示）是否有为障碍物来对  $c(n)$  进行赋值。如果有，则将该节点赋为无穷大，否则赋为零。

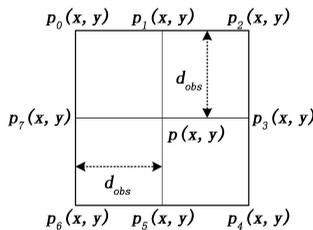


图 5 障碍物监测点示意图

为了测试本文的车身轮廓代价，本文参照文献 [8] 的方法在智能车的简化模型上加入碰撞检测点，如图 9 所示。同时，采用文献 [8] 的方法对扩展节点进行打分判断，并在图 7 的测试点上进行测试分析（ $L=5.555$  米表示车长， $D$  表示车宽）。

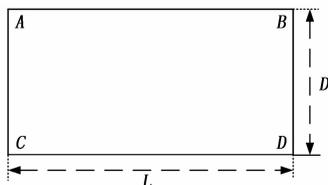
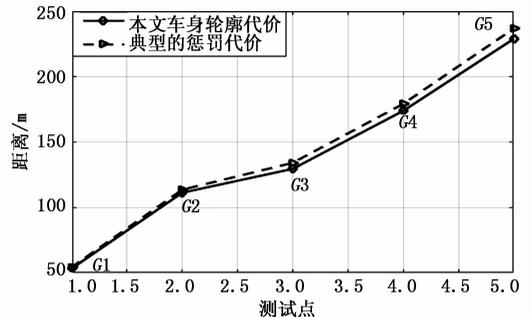


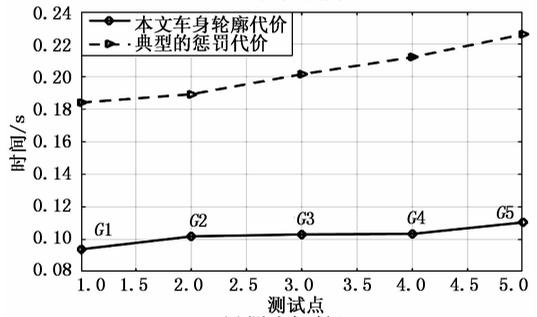
图 6 智能车简化模型的碰撞检测点

图 7 中的坐标点为所测试的起始点和目标点，其中 S 表示起始点，分别选取 G1, G2, G3, G4, G5 作为目标点。

在图 8 (a) 中， $x$  轴表示在图 7 所示的测试点， $y$  轴表示规划出路径的距离；在图 8 (b) 中， $x$  轴表示在图 7 所示的测试点， $y$  轴表示规划出路径的时间。同时，通过图 8



(a) 测试点距离



(b) 测试点时间

图 8 两种不同的方法在测试点的测试对比

的 (a) 和 (b) 所示，采用车身轮廓代价和典型的惩罚代价所规划的路径和时间对比可知：两种方法所规划的距离相差不大，但是所用的时间存在的一定差距。相比使用典型的惩罚代价，通过使用本文的车身轮廓代价能够减少算法运行的时间。此外，虽然加入车身轮廓代价的 A\* 算法所规划的距离有一定的增加，但是所用时间比传统 A\* 算法所用的时间少，这说明本文的车身轮廓代价是具有一定的优势的。

本文优化步骤如下：

- 1) 判断扩展节点四周的障碍物检测点是否为障碍物。
- 2) 如果是，则在扩展节点上增加障碍物标志位。否则，进行执行 3)。
- 3) 判断扩展节点是否为目标点。如果否，则执行步骤 1)；如果是则形成路径。

### 3.3 障碍物距离代价 $o(n)$

由于定位和地图存在着一定的误差，所以必须保证规划的路径与障碍物之间的距离尽量较大。通过在 A\* 算法中加入车身轮廓代价可以使得 A\* 算法的扩展节点约束到

图 9 的虚线内 (本文的  $d_{obs} = \frac{D}{2}$ )。但是对于智能车来说, 最优的路径应当靠近道路中心, 这样才能使得智能车在行驶过程中更为安全。

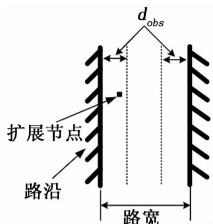


图 9 道路示意图

为了进一步使规划的路径更为合理和可靠, 本文采用 Voronoi 地图对规划所需要的栅格地图进行处理<sup>[10]</sup>。其中, Voronoi 地图是距离地图和广义泰森多边形相结合组成。首先, 通过 Voronoi 地图获得扩展节点到达障碍物的最近距离  $d_{min}$ 。然后, 将该距离作为约束条件带入到代价函数中, 如式 8 所示:

$$o(n) = \delta * \frac{1}{d_{min}} \quad (8)$$

在式 (8) 中,  $\delta$  表示障碍物安全距离代价的常数。 $d_{min}$  取倒数是因为 A\* 算法是选择代价值最小的节点作为扩展节点, 距离障碍物越远的节点对于智能车来说应该是越安全, 因此应该优先考虑该节点。

图 10 为加入障碍物距离的 A\* 算法和传统 A\* 算法在图 7 上的测试结果。其中, 在图 10 (a) 中, x 轴表示在图 7 所示的测试点, y 轴表示规划出路径的距离; 在图 10 (b) 中, x 轴表示图 7 中的测试点, y 轴表示规划出路径的时间。通过图 10 (a) 可以看出加入障碍物距离后的 A\* 算法和传统 A\* 算法在测试点所规划的路径距离大致相等。在 G1, G2 所测试的时间差距不大, 但是在后面的测试结果中传统 A\* 算法所用的时间与改进后的 A\* 算法所用时间差距越来越大, 如图 10 (b) 所示。

改进 A\* 算法流程如图 11 所示。

### 3.4 路径优化

由于车辆在转向中存在最小转弯半径, 不能像麦克纳姆轮一样全向移动。而传统的 A\* 算法所规划出的路径转折数目太多且不够平滑, 使得车辆在贴合路径时, 摇摆不定。所以必须对规划出来的路径进行进一步优化, 使其路径更加适用于车辆的运动特性。优化步骤如下:

1) 首先获得改进 A\* 算法的路径 path1, 并删出冗余点<sup>[11]</sup>, 得到转折点。

2) 依次从 path1 中取 pose [i-2], pose [i-1], pose [i] 进行拟合  $y(x)^1 = a * x^2 + b * x + c$ 。同时, 判断提取出来的转折点中, 是否包含终点。如果含有终点, 保存转折点到 path2, 结束对转折点优化的过程。否则, 执行下一步。

3) 以 pose [i-2]<sub>x</sub> 到 pose [i]<sub>x</sub> 为  $\Delta x$  的范围, 将  $\Delta x$  带

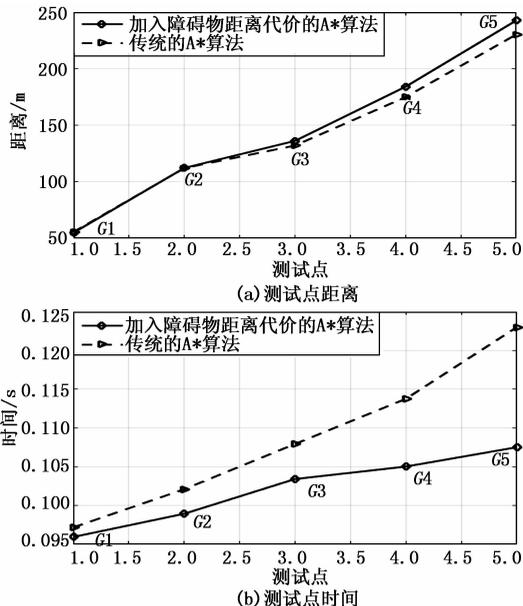


图 10 两种不同的方法在测试点的测试对比

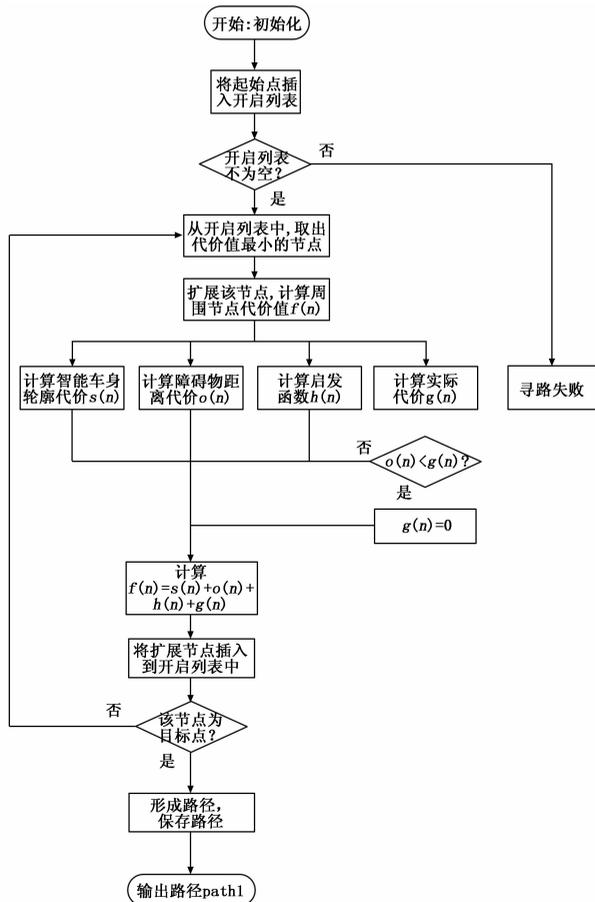


图 11 改进 A\* 算法流程图

入  $y(x)^1$  中求其一阶导数, 二阶导数, 带入公式 (9) 中, 并判断是否满足条件。如果满足, 则保存转折点到 path2, 跳出路径优化函数。否则, 求出  $y(x)^1$  的对称轴  $x = \frac{-b}{2 * a}$ ,

判断所提取的出来的转折点是否大于  $x = \frac{-b}{2 * a}$ 。同时，通过判断  $a$  的正负号确定  $y(x)^1$  的开口朝向。如  $a > 0$  且  $\text{pose}[i-2]_x > \frac{-b}{2 * a}$  则将  $\text{pose}[i-1]$  向左上角移动一个栅格。如果  $a > 0$  且  $\text{pose}[i]_x < \frac{-b}{2 * a}$ ，则将  $\text{pose}[i-1]$  向右上角移动一个栅格。如果  $a < 0$  且  $\text{pose}[i-2]_x > \frac{-b}{2 * a}$ ，则将  $\text{pose}[i-1]$  向左下角移动一个栅格。如果  $a < 0$  且  $\text{pose}[i]_x < \frac{-b}{2 * a}$ ，则将  $\text{pose}[i-1]$  向右下角移动一个栅格如公式 (11)。在有限次移动中，计算出每次移动的曲率带入公式 (10)。如果满足公式 (10)，则保留转折点到 path2 中。否则，移动  $\text{pose}[i-1]$ ，更新最小曲率的转折点并保留到 path2 中。(如图 12 和图 13 所示)。

4) 重复步骤 2)。

$$K = \left| \frac{(1 + (y')^2)^{3/2}}{y''} \right| \quad (9)$$

$$K < K_{\max} \quad (10)$$

$$\text{index}_{\text{new}} =$$

$$\begin{cases} \text{index} + nx - 1 & (a > 0 \text{ and } \text{pose}[i-2]_x > \frac{-b}{2 * a}) \\ \text{index} + nx + 1 & (a > 0 \text{ and } \text{pose}[i]_x < \frac{-b}{2 * a}) \\ \text{index} - nx - 1 & (a < 0 \text{ and } \text{pose}[i-2]_x > \frac{-b}{2 * a}) \\ \text{index} - nx + 1 & (a < 0 \text{ and } \text{pose}[i]_x < \frac{-b}{2 * a}) \end{cases} \quad (11)$$

$\text{index}_{\text{new}}$  是移动后节点的索引， $\text{index}$  是此刻节点的索引， $nx$  表示地图的宽度。(如图 12 所示)。

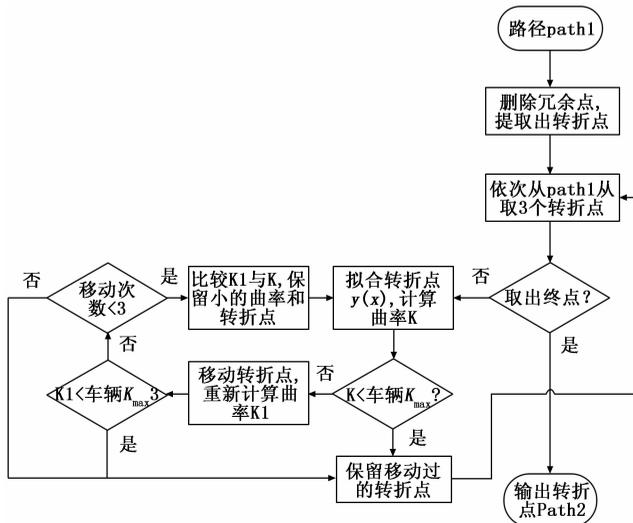


图 12 改进 A\* 算法所规划路径后优化流程图

$A, B, C$  为转折点;  $B'$  为移动后的转折点; 曲线  $\widehat{ABC}$  为转折移动之前所拟合的曲线; 虚线  $\widehat{A'B'C}$  为移动之后所拟

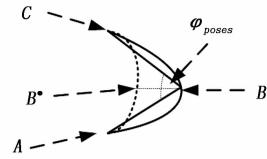


图 13 改进 A\* 算法转折点移动示意图之一

合的曲线;  $\varphi_{\text{poses}}$  为  $A, B, C$  所形成的角度。

### 3.5 路径平滑

为了使所规划路径更为平滑，有将转折路径用圆弧所替代已形成光滑路径<sup>[12]</sup>，也有用贝塞尔曲线或者样条曲线等方法对路径点优化的。本文采用贝塞尔曲线<sup>[13]</sup>对改进 A\* 算法的路径进行平滑处理。其中，贝塞尔曲线是法国工程师皮埃尔·贝塞尔在 1962 年提出，其形状可由曲线的控制点确定。因此，将改进改进 A\* 算法所规划的路径点作为贝塞尔曲线的输入，消除路径不平滑的情况。

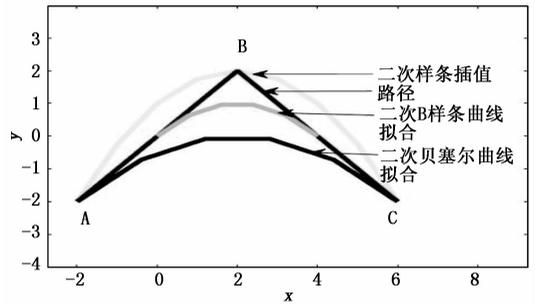


图 14 使用贝塞尔曲线示意图

使用不同拟合方法示意图如图 14 所示，其中  $A$  点， $B$  点和  $C$  点表示改进 A\* 算法的路径点。从图 14 可以看出，使用二次样条插值所生成的路径的曲率较大，二次 B 样条曲线拟合的起始点是  $A$  点和  $B$  点连线的中点，目标是  $B$  点和  $C$  点连线的中点，导致所生成的路径在起始点和目标点处不连续，而使用二次贝塞尔曲线拟合能够较好解决上述二次样条插值和二次 B 样条曲线拟合所带来的问题。

$$B(t) = \sum_{i=0}^n p_i (1-t)^{n-i} t^i \quad (12)$$

$$B(t) = (1-t)^2 * P_0 + 2 * t(1-t) * P_1 + t^2 * P_2 \quad t \in [0,1] \quad (13)$$

公式 (12) 为贝塞尔曲线的一般方程，公式 (13) 是本文所使用的二阶贝塞尔曲线， $P_0, P_1, P_2$  为二阶贝塞尔曲线的控制点。

### 4 算法验证

为了验证改进后的 A\* 算法的有效性，本文分别将 Dijkstra 算法、A\* 算法 (使用 Manhattan 距离)、A\* 算法 (使用 Euclidean 距离) 和改进 A\* 算法应用于成都信息工程大学校园的栅格地图 ( $4846 \times 2816$ ) 上，如图 15 所示，栅格大小为  $0.298 \text{ m/pixel}$ ，选择不同的地点进行实现。同时，本实验基于 ROS (开源机器人操作系统) 平台，并使用 Rviz 进行可视化。计算机配置为：Ubuntu16.04 LTS，

处理器 Intel i5-6500, 主频为 3.2 Ghz, 运行内存为 8 G。

本次实验通过在地图上选择起始点和终点, 上述 4 种算法进行路径规划。通过比较规划路径的长度、与障碍物的距离、路径最大转角和遍历的节点数等方式来进行判断路径的优劣。对区域一、区域二和区域三进行路径规划, 规划结果见图 16、图 17 和图 18。表 1、表 2 和表 3 分别给出了不同区域规划的指标对比。

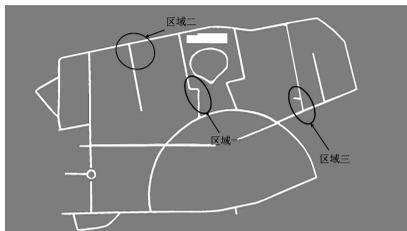


图 15 学校的校园地图

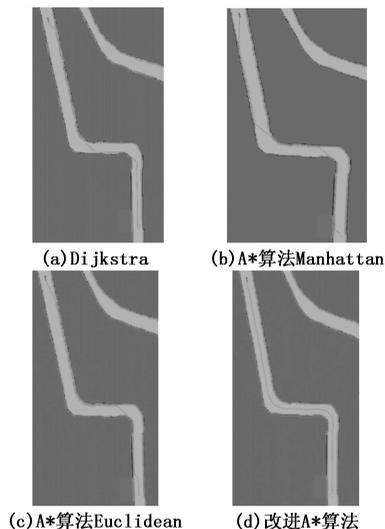


图 16 在区域一测试的结果

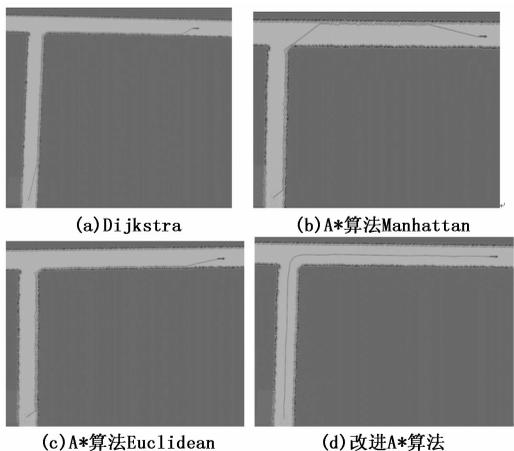


图 17 在区域二测试的结果

从图 16、图 17 和图 18 中可以看出, 改进的 A\* 算法所规划出来的路径较为平滑, 具有连续的曲率并能够较好地满足车辆运动学模型。同时, Dijkstra 算法、A\* 算法

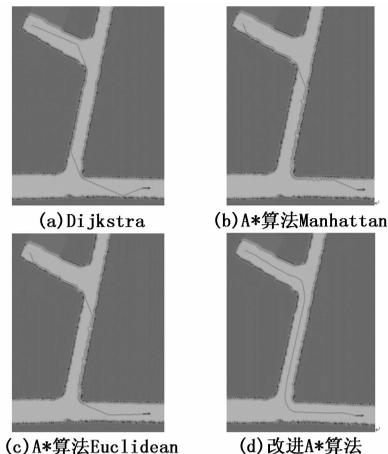


图 18 在区域三测试的结果

(使用 Manhattan 距离)、A\* 算法 (使用 Euclidean 距离) 在测试区域测试中都存在起始时刻路径转折角度过大, 路径贴合路沿, 路径不平滑和转折次数较多等问题。在加入车身轮廓代价、障碍物距离代价和对路径优化过后能够较好地解决这些问题 (如图 16、17 和 18 的 (d) 图所示)。这说明改进后的 A\* 算法能够更好地解决传统 A\* 算法路径规划所存在的问题。

表 1 区域一测试对比

	Dijkstra 算法	A* 算法 Manhattan	A* 算法 Euclidean	改进 A* 算法
总时间/s	0.110 978	0.100 396	0.103 153	0.096 495
路径距离/m	139.208 5	147.054 4	142.247 2	142.880 8
遍历节点数	23 184	1 548	8 191	1 519
路沿最近距离/m	0	0	0	2.061 32
最大转角/(°)	50	46	48	18
平滑时间/s	0	0	0	0.000 102
转折数目	4	3	3	0

表 2 区域二测试对比

	Dijkstra 算法	A* 算法 Manhattan	A* 算法 Euclidean	改进 A* 算法
总时间/s	0.118 799	0.099 184	0.106 390	0.060 559
路径距离/m	122.939 5	140.202 8	137.215 9	127.632 0
遍历节点数	24 992	4 332	9 879	1 520
路沿最近距离/m	0	0	0	2.297 96
最大转角/(°)	37	84.9	58	7.8
平滑时间/s	0	0	0	0.000 076
转折数目	3	3	1	0

由表 1、表 2 和表 3 中能够看出改进后的 A\* 算法规划时间小于 Dijkstra 算法、A\* 算法 (使用 Manhattan 距离)、A\* 算法 (使用 Euclidean 距离) 的规划时间。同时, 所遍历的节点数最少, 并且与路沿的最近距离都大于车身宽度

表 3 区域三测试对比

	Dijkstra 算法	A * 算法 Manhattan	A * 算法 Euclidean	改进 A * 算法
总时间/s	0.102 936	0.097 943	0.100 858	0.062 720
路径距离/m	88.369 1	92.842 003	92.562 3	89.933 2
遍历节点数	9 766	2 639	4 811	1 043
路沿最近距离/m	0	0	0	1.651 44
最大转角/(°)	46	50	47	16
平滑时间/s	0	0	0	0.000 077
转折数目	5	4	2	0

的一半。此外，改进 A \* 算法所生成路径的最大转角都是小于车辆最大转向角。因此，通过该方法所生成的路径是能够被智能车所执行的。

### 5 结束语

本文针对传统 A \* 算法所规划的路径转折角度较大，路径不平滑，所规划的路径贴合障碍物和未考虑车辆自身约束条件等问题进行改进，得到如下结论：

- 1) 通过建立车辆运动学模型得到约束条件将其带入启发函数中使得节点代价值更为合理。
- 2) 利用车身轮廓代价去掉 A \* 算法所存在不合理的扩展节点，进而减少 A \* 算法所遍历的节点数目。
- 3) 通过加入障碍物距离代价解决路径靠近障碍物的问题，使得所规划的路径更为合理，更为安全。
- 4) 使用贝塞尔曲线对路径转折点进行拟合使所规划的路径更平滑，更适用于智能车的运动特性。
- 5) 由于本文算法是主要针对于在实验阶段中的 AGV 或者智能车，并且所实验的载体是在中低速的环境中运行。其次，将该算法进行改进并用于高速行驶的移动载体是后续研究的重点。

### 参考文献：

[1] 孟海华, 江洪波, 汤天波. 全球自动驾驶发展现状与趋势 (上) [J]. 华东科技, 2014 (9): 66-68.

[6] Jobson J B, Woodell G A. Multi-scale retinex for color restoration on graphics processing unit [J]. Journal of Real-Time Image Processing, 2014, 10 (2): 239-253.

[7] 张弛, 谭南林. 基于改进型 Retinex 算法的雾天图像增强技术 [J]. 北京航空航天大学学报, 2019, 45 (2): 309-316.

[8] Liu H J, Sun X K. Low-light video image enhancement based on multiscale Retinex-like algorithm [A]. Chinese Control and Decision Conference [C]. 2016.

[9] Land E H. Recent advances in retinex theory and some implications for cortical computations: color vision and the natural image [J]. Proceedings of the National Academy of Sciences of the United States of America, 1983, 80 (16): 5163.

[10] 谢凤英, 汤萌, 等. 基于 Retinex 的图像增强方法综述 [J]. 数据采集与处理, 2019, 34 (1): 1-11.

[2] 杨瑶, 付克昌, 蒋涛, 等. 启发式 RRT 算法的 AGV 路径规划 [J/OL]. 计算机工程与应用: 1-10 [2019-12-31].

[3] 辛煜, 梁华为, 杜明博, 等. 一种可搜索无限个邻域的改进 A \* 算法 [J]. 机器人, 2014, 36 (5): 627-633.

[4] Huang Y, Ding H, Zhang Y, et al. A motion planning and tracking framework for autonomous vehicles based on artificial potential field-elaborated resistance network (APFE-RN) approach [J]. IEEE Transactions on Industrial Electronics, 2019.

[5] 周兰凤, 杨丽娜, 方华. 基于滑移预测的月球车路径规划研究 [J]. 计算机应用与软件, 2019, 36 (11): 47-50.

[6] 王殿君. 基于改进 A \* 算法的室内移动机器人路径规划 [J]. 清华大学学报: 自然科学版, 2012, 52 (8): 1085-1089.

[7] 陈伟华, 林颖, 文宗明, 等. 基于双重 A~ \* 算法的移动机器人动态环境路径规划 [J]. 组合机床与自动化加工技术, 2018 (4): 127-130.

[8] 马飞, 杨峰岫, 顾青, 等. 基于改进 A \* 算法的地下无人铲运机导航路径规划 [J]. 农业机械学报, 2015, 46 (7): 303-309.

[9] 徐娜, 陈雄, 孔庆生, 等. 非完整约束下的机器人运动规划算法 [J]. 机器人, 2011, 33 (6): 666-672.

[10] 杜明博, 梅涛, 陈佳佳, 等. 复杂环境下基于 RRT 的智能车辆运动规划算法 [J]. 机器人, 2015, 37 (4): 443-450.

[11] Ayawli B B K, Mei X, Shen M, et al. Mobile robot Path planning in dynamic environment Using Voronoi diagram and computation geometry technique [J]. IEEE Access, 2019, 7: 86026-86040.

[11] 程传奇, 郝向阳, 李建胜, 等. 融合改进 A~ \* 算法和动态窗口法的全局动态路径规划 [J]. 西安交通大学学报, 2017, 51 (11): 137-143.

[12] 陆皖麟, 雷景森, 邵炎. 基于改进 A~ \* 算法的移动机器人路径规划 [J]. 兵器装备工程学报, 2019, 40 (4): 197-201.

[13] Ding H, Li Y, Chai Y, et al. Path Planning for 2-DOF manipulator based on Bezier curve and A \* algorithm [A]. 2018 Chinese Automation Congress (CAC) [C]. IEEE, 2018: 670-674.

[11] Guo X J, Li Y. LIME: Low-light Image enhancement via illumination map estimation [J]. IET Image Processing, 2017, 26 (2): 982-993.

[12] He K, Sun J, Tang X. Guided image filtering [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2013, 35 (6): 1397-1409.

[13] 芦碧波, 李玉静. 视觉自适应多尺度对数色调映射算法 [J]. 小型微型计算机系统, 2017, 38 (3): 625-629.

[14] 沈薛晨, 刘钧, 高明. 基于小波-Contourlet 变换的偏振图像融合算法 [J]. 红外技术, 2020, 42 (2): 182-189.

[15] 荣传振, 贾永兴, 杨宇. 增强融合图像视觉效果的图像融合方法 [J]. 信号处理, 2019, 35 (3): 327-333.

[16] Shin Y, Jeong S, Lee S. Efficient naturalness restoration for non-uniform illumination images [J]. IET Image Processing, 2015, 9 (8): 662-671.

(上接第 159 页)