

● 基础科学与技术 Basic Science & Technology

一种基于改进混合 A* 的智能车路径规划算法

齐尧¹, 徐友春², 李华², 王任栋¹

(1. 陆军军事交通学院 学员五大队, 天津 300161;

2. 陆军军事交通学院 军事交通运输研究所, 天津 300161)

摘要:为给智能车规划一条安全、平滑且满足车辆非完整性约束的路径, 针对混合 A* 算法规划所得路径安全性不足的问题, 提出惩罚式图搜索方法, 能够得到更为安全、合理的路径。另外, 提出一种基于多目标约束的后处理方法以生成曲率连续变化的可行驶路径。通过仿真与实车实验, 验证了算法的时效性、实用性与可靠性。

关键词:路径规划; 智能车; 惩罚式搜索算法; 混合 A*

DOI:10.16807/j.cnki.12-1372/e.2018.08.020

中图分类号: TP24

文献标志码: A

文章编号: 1674-2192(2018)08-0085-06

Hybrid A* Algorithm-Based Path Planning for Intelligent Vehicle

QI Yao¹, XU Youchun², LI Hua², WANG Rendong¹

(1. Fifth Team of Cadets, Army Military Transportation University, Tianjin 300161, China;

2. Institute of Military Transportation, Army Military Transportation University, Tianjin 300161, China)

Abstract: To plan a safe, smooth and non-holonomy restricted path for an intelligent vehicle, the paper studies a graph search method with penalty to improve the safety of hybrid A* algorithm, and puts forward a post-processing method based on multi-object constraint to generate a drivable path with a continuous smooth curve. Both the simulation experiment and real running validate this method's timeliness, practicability and reliability.

Keywords: path planning; intelligent vehicle; penalty search algorithm; hybrid A*

智能汽车作为一种智能化的轮式移动机器人, 备受诸多行业和研究机构的关注, 目前国内外已有特定场景下的无人驾驶汽车投入试运营。路径规划是轮式机器人的关键技术之一, 是确保行驶安全、缩短行驶距离、增加行驶舒适性的前提。路径规划的任务是找到一条从当前点到目标点的局部行驶路径, 使智能车在本次任务中能安全快速地到达目标位置^[1]。

机器人路径规划算法主要包括基于采样

(RRT、PRM)的方法、轨迹直接生成法、基于图搜索(A*、D*)的方法^[2], 另有传统算法如遗传算法、蚁群算法、模拟退火算法等。基于采样的方法具备搜索速度快、不需要对环境进行建模的优点, 但其随机采样的特性导致搜索速度慢、路径不平滑; 轨迹生成算法主要有直接构造法与路径—速度分解法两类, 其中路径—速度分解法将路径与速度生成分开, 是使用较广的方法^[4], 但这类方法在非结构化复杂环境下适应性较差, 对此许多学者提出将此类

收稿日期: 2018-03-27; 修回日期: 2018-04-03.

作者简介: 齐尧(1994—), 男, 硕士研究生;

徐友春(1972—), 男, 博士, 教授, 博士研究生导师。

方法所用的曲线用于对搜索得到的初始路径进行平滑、插值^[3-6]。基于图搜索的算法具有最优性和实时性,但传统图搜索算法得到的路径不能够满足车辆非完整性约束。针对这一问题,Dolgov等^[7-8]提出基于A*的改进算法。Dolgov等^[7]提出一种满足车辆运动学特性的混合A* (hybrid A*) 路径搜索算法,并采用数值优化方法对路径进行后处理,能够规划出接近全局最优的可行路径,但不能保证车辆在起始位置和目标位置保持曲率连续变化;Chu等^[5]使用混合A*算法作为路径搜索方法、PH三次曲线作为路径后处理方法,但是路径不能够与障碍保持足够的安全距离;Petereit等^[9]将混合A*算法较好地应用到室外非结构化环境中,并在此基础上增加途经点规划算法;文献^[10]使用A*搜索结果与Reeds-Shepp曲线共同作为启发式数值,并使用Reeds-Shepp曲线进行节点扩展,加快了搜索的速度,但容易导致路径中频繁出现倒车、变换行驶方向等不合理的情况出现。

针对现有方法存在的问题,本文提出一种基于混合A*的路径规划算法,算法的流程如图1所示。首先使用本文提出的惩罚式Reeds-Shepp曲线与二维规划结果作为启发式函数,结合车辆非完整性约束进行三维规划,并在规划的过程中引入合理的惩罚函数,极大地提高了初始规划路径的安全性和合理性;然后建立多目标约束函数,通过共轭梯度下降的方法对路径进行最优化处理,得到曲率连续变化的路径,增加行驶舒适性。通过多种环境下的仿真与实车实验,验证了算法的实时性、有效性和可靠性,算法能够较好地解决非结构化复杂环境下的智能车路径规划问题。

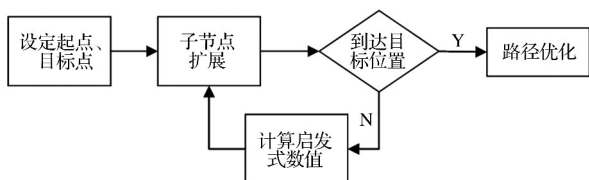


图1 算法流程

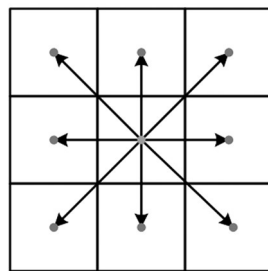
1 基于图搜索的混合A*算法

基于搜索的规划算法将障碍物与搜索区域划分为离散的栅格空间,在图结构中进行搜索。诸如A*、D*等最优搜索算法的目标是在给定起始点与目标点之间搜索一条全局最优的路径,但该路径不

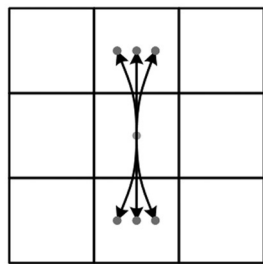
能满足车辆的非完整性约束。混合A*算法将车辆的运动学约束融合在传统A*算法中,节点与节点间以连续的三维运动学状态连接,路径满足车辆非完整性约束。

1.1 子节点扩展方式

算法采用空间连续的子节点扩展方式(如图2所示),子节点位置根据车辆最小转弯半径以及节点长度确定。车辆状态空间为 (x, y, θ, δ) ,其中 (x, y, θ) 为车辆位姿状态, $\delta=1$ 为车辆正向行驶状态, $\delta=-1$ 为车辆倒车状态。通过该扩展方式能够使规划的路径满足运动学约束。图2中心点表示父节点,周围点表示子节点,图2(a)中A*算法的子节点数量为8个且位于栅格中心,图2(b)中混合A*算法子节点数量为6个且位于栅格中的任意位置且满足车辆非完整性约束。



(a) A*算法



(b) 混合A*算法(hybrid-A*)

图2 A*与混合A*算法子节点

1.2 启发式函数

启发式函数(Heuristics)是影响A*类图搜索算法时效性、结果优异性的关键因素之一。为满足车辆运动学约束与环境约束,算法采用两种启发式并行的方法,并选取其中较大的值作为搜索的启发式数值。

第1种启发式函数是在忽略环境障碍、考虑车辆非完整性约束的条件下得到,其数值是从节点 $(x_{node}, y_{node}, \theta_{node})$ 到达目标位置 $(x_{goal}, y_{goal}, \theta_{goal})$,且满足非完整性约束下的最短路径,最短路径可以根据状态晶格或Reeds-Shepp曲线进行获取,由于状态晶

格对内存消耗大,本文采用 Reeds-Shepp 曲线对第 1 种启发式数值进行计算。通过离线计算(0,0,0)到达任意点 (x,y,θ) 的距离,实时通过三维向量旋转与平移得到任意节点 $(x_{\text{node}},y_{\text{node}},\theta_{\text{node}})$ 到达目标位置 $(x_{\text{goal}},y_{\text{goal}},\theta_{\text{goal}})$ 的启发式数值。该启发式保证了搜索过程中子节点扩展时受到终点姿态的约束。

第 2 种启发式函数在忽略车辆非完整性约束、考虑环境约束的条件下得到。计算这类启发式数值是在二维空间,可以使用 A*、Dijkstra、动态规划等 3 种方式进行计算。本文根据环境中障碍数量选择动态规划或 A* 算法得到本项启发式数值。计算该启发式时需在三维位形空间内进行碰撞检测,即 $f(x,y,\theta)$ 空间。 $f(x,y)$ 内存在一个角度使其不发生碰撞,则认为 (x,y) 是无碰撞的点。即

$$f(x,y)=\exists_{\theta} f(x,y,\theta) \quad (1)$$

选择两者中较大的值作为启发式:

$$h(s,s_{\text{goal}})=\max(h_1(s,s_{\text{goal}}),h_2(s,s_{\text{goal}})) \quad (2)$$

式中: h_1 为在忽略环境障碍、考虑车辆非完整性约束的条件下得到的启发式数值; h_2 为在忽略车辆非完整性约束、考虑环境约束的条件下得到的启发式数值。

2 惩罚式混合 A* 算法

在智能驾驶汽车行驶的过程中,根据应用场景不同对车辆倒车行驶的要求也不同。比如,在高速道路上不允许倒车,而在停车场内允许倒车。本文提出在启发式数值计算和子节点扩展的过程引入惩罚函数,根据决策需求控制车辆能否出现倒车以及出现倒车后的惩罚系数的大小,使路径更适应于不同场景内车辆行驶需求;同时对路径增加风险评估,使路径与障碍保持安全距离,增加行驶安全性。

2.1 路径惩罚函数

从节点 $node_i$ 到达 $node_{i+1}$ 的消耗值计算式为

$$c(node_i,node_{i+1})=d(node_i,node_{i+1}) \cdot \left(1+\frac{|\delta_{i+1}-1|}{2} \cdot \omega_r + \frac{|\delta_{i+1}-\delta_i|}{2} \cdot \omega_t\right) + risk_{i+1} \quad (3)$$

式中: $d(node_i,node_{i+1})$ 为两个节点间的欧式距离; $risk_{i+1}$ 为第 $i+1$ 个节点与障碍距离的风险惩罚值; δ 为车辆行驶状态参数, $\delta=1$ 为车辆处于正向行驶的状态, $\delta=-1$ 为车辆处于倒车行驶的状态; ω_r 为倒车状态下行驶的惩罚系数; ω_t 为车辆行驶方向改变的惩罚系数。

假设距离节点 i 最近的障碍点为 O_i ,其中 d_{obstacle} 为 O_i 与节点 i 的距离, ω_o 为距离障碍小于安全距离 d_{max} 时的惩罚系数。则障碍距离的惩罚值 $risk_i$ 为

$$risk_i = \begin{cases} \omega_o \cdot (d_{\text{max}} - d_{\text{obstacle}}), & d_{\text{obstacle}} \leq d_{\text{max}} \\ 0, & \text{otherwise} \end{cases} \quad (4)$$

式(3)中的惩罚函数即保证了规划路径的过程中能够与障碍保持安全距离,当可行驶区域较为狭窄时,该惩罚值也不会阻碍规划。式(3)中的 ω_r 、 ω_t 数值均由决策方案要求灵活决定,如允许倒车但不希望频繁倒车时,可以相应增加 ω_r 、 ω_t 的数值;如不允许倒车可以增加 ω_r 、 ω_t 至无穷大,该惩罚函数的设置能够使路径规划算法满足不同要求下的行驶要求。

2.2 惩罚式 Reeds-Shepp 曲线

Reeds-Shepp 曲线作为本文搜索算法的启发式计算方式和子节点扩展方式之一,对路径产生较大的影响。其原本的计算方式是在设定的集合中选择最短的路径,选择的依据是路径的欧式长度^[11]。本文提出惩罚式 Reeds-Shepp 曲线,即在 Reeds-Shepp 曲线选择最优路径时对倒车状态下行驶的距离、变换行驶方向的次数增加惩罚,在计算 Reeds-Shepp 距离时对倒车状态下的行驶距离乘上惩罚系数 ω_r ,每变换一次行驶方向也增加惩罚数值 ω_t ,得到惩罚式 Reeds-Shepp 曲线。根据惩罚系数的不同,可以得到不同类型的最优曲线:当 $\omega_r=1$ 、 $\omega_t=0$ 时,得到传统的 Reeds-Shepp 曲线;当 $\omega_r>1$ 、 $\omega_t>0$ 时,可以避免车辆为减小行驶路径长度而频繁倒车、变换行驶方向的问题;而当数值为无穷时,则不会出现倒车的情况。算法在启发式数值的计算与 Reeds-Shepp 曲线扩展时均采用本文所提惩罚式曲线,惩罚系数与式(3)中的系数数值相同。

3 路径后处理方法

根据以上方法生成的路径是可行驶的,但当节点扩展时,子节点为最小转弯半径下的弧线,且扩展时使用了 Reeds-Shepp 曲线,故路径各点的曲率总为极值。这种路径很难被执行,同时会降低行驶舒适性、增大车辆机械磨损。本文对路径中各点的曲率、平滑度、与障碍的距离进行约束,建立多目标函数,采用共轭梯度下降的方法对目标函数求极值以得到较为平滑的路径。通过 2.1 中的惩罚式搜索

算法生成的路径由一系列节点 $X_i = (x_i, y_i)$, $i \in [0, N]$ 组成; 设 $\Delta X_i = X_i - X_{i-1}$ 表示两个节点组成的矢量; O_i 表示距离节点 X_i 最近的障碍点的位置。

建立的目标函数为

$$\omega_{ob} \sum_{i=0}^N (|X_i - O_i| - d_{\max})^2 + \omega_s \sum_{i=0}^{N-1} (\Delta X_{i+1} - \Delta X_i)^2 + \omega_k \sum_{i=1}^{N-1} (k_i - k_{\max})^2 \quad (5)$$

式中 ω_{ob} 、 ω_s 、 ω_k 为各项所占权重, 权重均由多次实验的方式获取。

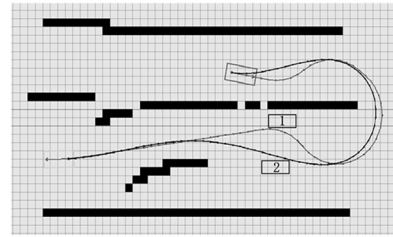
式(5)中第1项约束节点与障碍的安全距离, d_{\max} 为平滑时节点距离障碍点的最小安全距离, 大小为车辆后轴中心点距离车辆前方顶点的斜线长度, 当 $|X_i - O_i| > d_{\max}$ 时第1项产生作用, 否则梯度为0; 第2项对路径进行了平滑; 第3项对任意节点的曲率 k_i 进行约束, 曲率数值必须小于 k_{\max} , 最大曲率 k_{\max} 由车辆的运动学约束决定, 当 $k_i \leq k_{\max}$ 时, 第3项的梯度取值为0。式(5)各项梯度见式(6), 并采用共轭梯度下降的方法对式(5)求极值, 得到了较为平滑的路径。

$$\begin{cases} 2(|X_i - O_i| - d_{\max}) \frac{X_i - O_i}{|X_i - O_i|} & |X_i - O_i| > d_{\max} \\ 4(2X_i - X_{i+1} - X_{i-1}) & \\ 2(k_i - k_{\max}) \frac{\partial k_i}{\partial X_i} & k_i > k_{\max} \end{cases} \quad (6)$$

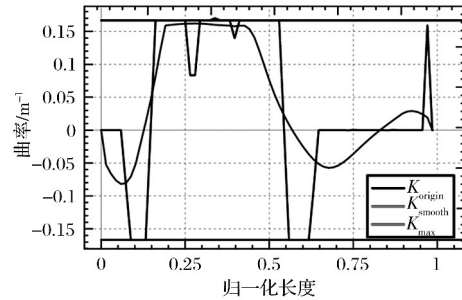
图3所示为平滑前后对比图, 在图3环境中, 图3(a)中1号和2号曲线分别表示搜索得到的初始路径和经过后处理优化得到的最终路径, 对路径长度归一化处理并求得的曲率如图3(b)所示, 经过平滑后的路径曲率较为连续地变化, 更满足车辆行驶的要求, 能够有效减少车辆抖动。

4 实验与分析

通过仿真与实车实验, 对本文提出的惩罚式混合 A* 算法的可靠性、有效性与实用性进行验证。通过搭建仿真环境模拟车辆行驶过程, 在多种环境进行实验验证算法的可靠性、优异性与时效性; 通过与传统混合 A* 算法进行比较, 对本文的惩罚式搜索算法的效果进行了对比验证, 验证了算法的有效性; 最后通过实车实验, 验证了算法的实用性。



(a) 路径对比



(b) 曲率对比

图3 平滑前后对比

4.1 仿真实验

通过搭建环境仿真平台, 使用矩形框模拟车辆几何模型, 根据可任意设定的起始和目标位置、朝向与曲率, 进行规划算法的仿真实验。设定模拟车辆的最小转弯半径为 6 m, 最大曲率为 0.167 m^{-1} , 实验只涉及智能车路径规划, 并未进行速度规划, 以路径长度、曲率变化率、变换行驶方向的频率作为评估路径优越性的标准。

4.1.1 规划实验

为验证算法的时效性与适应性, 本文构建仿真环境, 模拟智能车的环境感知、路径规划过程, 进行了多次规划。图4(a)为初始时车辆感知得到的障碍环境, 车辆行驶任务为从起始位置(左下角)按照规划路径行驶至目标位置(右上角), 随着车辆不断前进, 记录已走过的环境并将新环境加入地图中, 如果规划所得路径被障碍所阻断, 则进行重新规划。图4(b)为行驶途中的规划结果。图4(c)为到达目标位置所走过的路径, 本次实验共计规划 42 次, 规划成功的次数为 42 次。图4(d)为初始规划和后处理的总时间, 其中最短时间为 10 ms, 最长为 308 ms, 平均时间为 52.3 ms。仅有的一次时长超过 150 ms 的规划结果(如图4(b)所示), 路径较为复杂, 在此场景下对规划时间的要求可适当放宽, 总体时间满足智能车实时规划的要求。

4.1.2 与初始搜索路径对比

为验证本文所提惩罚式函数的有效性, 将惩罚

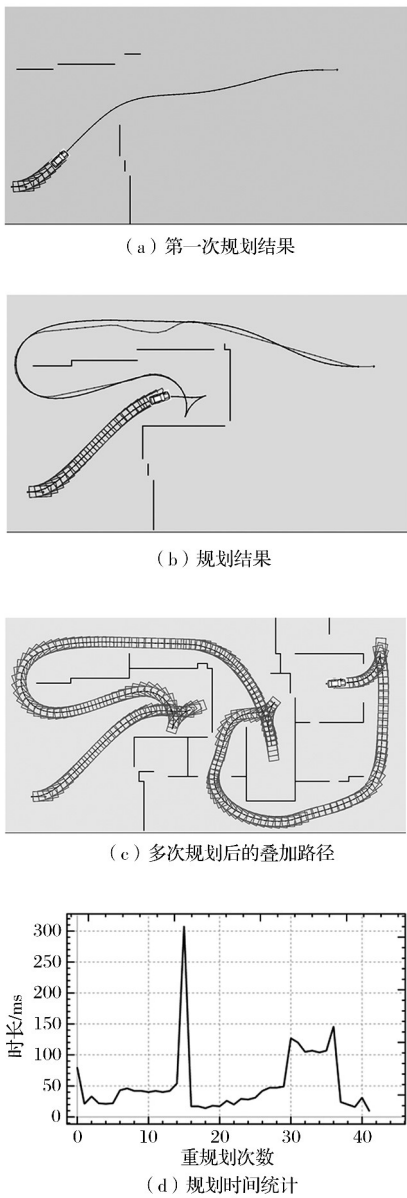


图 4 仿真实验结果

式混合 A* 算法与文献[10]中混合 A* 算法进行实验对比。在图 5 所示环境中分别进行实验。

惩罚函数中的惩罚系数 $\omega_r=2$ 、 $\omega_l=10$ ，根据表 1 中的数据对比得到：图 5(a) 中搜索结果路径长度虽然短，但是倒车状态下行驶的距离很长，路径中频繁出现变换行驶方向的情况，路径不够合理。同时搜索所得路径距离障碍较近，安全性不够强。图 5(b) 为本文提出的惩罚式搜索算法的结果，虽然路径长度较长，但是按照本文惩罚式搜索方式计算的路径长度开销比传统方法的开销小许多，使得路径更为合理并有利于行驶，安全性更高。

4.2 实车实验

通过“JJUV-6”平台进行实验，验证算法的实用

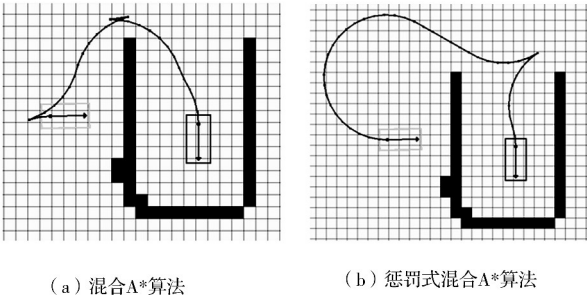


图 5 初始搜索对比结果

表 1 仿真实验数据对比

参数	路径长度/m	倒车状态行驶距离/m	变换行驶方向次数	路径长度开销	子节点数量	搜索时间/ms
传统方法	27	24	3	81	149	20
本文方法	46	10	1	66	286	27

性，平台的部分参数见表 2。根据智能车 64 线激光雷达的检测范围，本文规划时使用的栅格地图大小为 100 m×100 m，栅格中障碍分辨率为 25 cm，角度分辨率为 5°。在实际复杂环境下，任务为从起点位置行驶至目标位置。图 6(a) 中 1 号框为车辆起始位置，2 号框为目标位置；图 6(b) 为车辆在起始位置时 64 线激光雷达的点云数据，左上方箭头与图 6(a) 中 1 号框位置相同，右下方箭头与图 6(a) 中 2 号框位置相同，箭头的方向即为车辆在该位置时的目标朝向。

表 2 JJUV-6 参数

最小转弯半径/m	最大曲率/(m ⁻¹)	轴距/m	平台尺寸(长×宽×高)/(mm×mm×mm)
6	0.167	2.915	4 906×1 975×2 294

规划时的搜索树、初始路径、最终路径分别如图 7 所示，图 7(a) 中起始位置附近曲线为搜索过程的子节点，曲线中靠近目标位置的一段表示 Reeds-Shepp 曲线扩展所得路径；图 7(b) 中曲线为经过后处理的结果。图 7 所示环境中搜索节点数为 182 个，初始搜索与平滑的总时间为 23 ms，同时得到该路径的曲率变化如图 7(c) 所示，各节点的曲率均小于最大曲率，同时曲率变化较为连续，满足车辆行驶要求和实时性要求，进一步验证了算法的实用性。

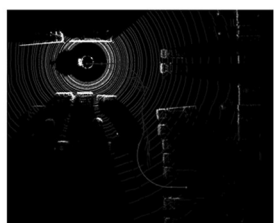


(a) 视频图像

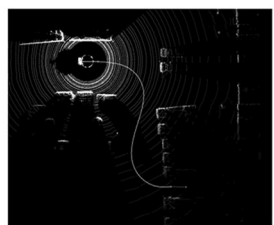


(b) 点云数据

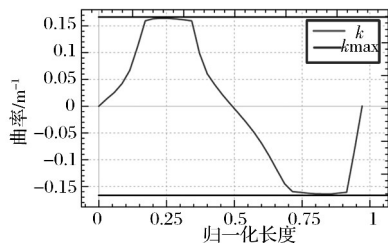
图6 实验场景



(a) 搜索结果



(b) 平滑结果



(c) 曲率变化曲线

图7 实车规划结果

5 结 语

智能车的路径规划算法既要保证规划所得路径可行驶以及与障碍点保持安全距离,还需要保证

路径足够平滑以行驶过程中的抖动。本文基于混合A*算法提出惩罚函数,在空旷的区域能够与障碍保持安全距离,根据决策要求可选择车辆能够倒车,并根据车辆状态和环境决定惩罚系数,能够得到不同需求的路径;采用共轭梯度下降法对路径进行后处理并得到了平滑的路径,增加了行驶舒适性。仿真实验与实车实验结果表明该算法能够有效解决智能车的规划问题。本文算法主要解决智能车在多障碍低速运动环境下的规划问题,下一步可解决高速运动环境中的规划问题。

参考文献:

- [1] LAVELLE S.M. Planning algorithms [M]. New York: Cambridge University Press, 2006:3-10.
- [2] 余卓平,李奕姗,熊璐.无人车运动规划算法综述[J].同济大学学报(自然科学版),2017,45(8):1150-1159.
- [3] 陆峰,章永进,李鹏,等.无人车导航路径关键点插值算法[J].军事交通学院学报,2016,18(2):89-93.
- [4] PHAM Q C, CARON S, LERTKULTANON P, et al. Admissible velocity propagation: beyond quasi-static path planning for high-dimensional robots[J]. Eprint Arxiv, 2014,36(1):1-8.
- [5] 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(4):653-668.
- [6] ELBANHAWI M, SIMIC M. Randomised kinodynamic motion planning for an autonomous vehicle in semi-structured agricultural areas [J]. Biosystems Engineering,2014, 126(39):30-44.
- [7] DOLGOV D, THRUN S, MONTEMERLO M, et al. Path planning for autonomous vehicles in unknown semi-structured environments [J]. International Journal of Robotics Research, 2010,29(5):485-501.
- [8] FERGUSON D, HOWARD T M, LIKHACHEV M. Motion planning in urban environments[J]. Journal of Field Robotics, 2010,25(11-12):939-960.
- [9] PETEREIT J, EMTER T, FREY C W, et al. Application of Hybrid A* to an autonomous mobile robot for path planning in unstructured outdoor environments[C]// German Conference on Robotics. 2012:1-6.
- [10] KURZER K. Path planning in unstructured environments: a real-time Hybrid A* implementation for fast and deterministic path generation for the KTH research concept vehicle [D]. Stockholm, Sweden: KTH Royal Institute of Technology, 2016.
- [11] REEDS J A, SHEPP L A. Optimal paths for a car that goes both forward and backward [J]. Pacific Journal of Mathematics, 1990,145(2):367-393.

(编辑:张峰)