

基于 A* 引导域的 RRT 智能车辆路径规划算法^①

冯来春^{1,2}, 梁华为², 杜明博², 余彪²

¹(中国科学技术大学 自动化系, 合肥 230027)

²(中国科学院合肥物质科学研究院 应用技术研究所, 合肥 230027)

摘要: 为了泛化 RRT(快速搜索随机树)算法在智能车辆路径规划领域内的应用, 解决该算法搜索效率低、最近邻搜索函数不合理等问题, 本文提出了一种基于 A* 引导域的 RRT 路径规划算法. 该算法将 A* 算法与 RRT 搜索算法进行有效地结合, 利用由 A* 算法在低分辨率栅格图中生成的最短路径来构建引导域, 以提升 RRT 算法的采样效率; 同时在设计 RRT 算法的最近邻搜索函数时考虑车辆自身约束, 以增强搜索树节点选择的合理性. 通过仿真实验和实车测试, 对该算法的优越性、有效性和实用性进行了验证.

关键词: RRT; 采样策略; A*; 智能车辆

引用格式: 冯来春, 梁华为, 杜明博, 余彪. 基于 A* 引导域的 RRT 智能车辆路径规划算法. 计算机系统应用, 2017, 26(8): 127-133. <http://www.c-s-a.org.cn/1003-3254/6023.html>

Guiding-Area RRT Path Planning Algorithm Based on A* for Intelligent Vehicle

FENG Lai-Chun^{1,2}, LIANG Hua-Wei², DU Ming-Bo², YU Biao²

¹(Department of Automation, University of Science and Technology of China, Hefei 230027, China)

²(Institute of Applied Technology, Hefei Institutes of Physical Science, Chinese Academy of Sciences, Hefei 230027, China)

Abstract: This paper proposes a RRT path planning algorithm based on the guiding-area which is generated with the A* algorithm. This algorithm can benefit the domain from the following aspects: the applications of RRT algorithm to the field of path planning for the intelligent vehicle can be improved significantly. The performance of the traditional RRT algorithm can be enhanced by solving some inherent issues, such as low searching efficiency, irrational nearest neighbour searching functions etc. The novel algorithm combines A* and RRT effectively. Based on low resolution grid map, A* algorithm is applied to construct the guiding area, which is used to improve the sampling efficiency. To enhance the reasonableness of the selection of searching tree node, the vehicle's constraints are considered in the design of the nearest neighbour searching function. Finally, the superiority, validity and practicability of the proposed algorithm is verified in simulations and experiments with the real vehicle

Key words: RRT; sampling strategy; A*; intelligent vehicle

1 引言

无人驾驶技术有助于改善车辆行驶的安全性和舒适性, 受到了学术界和工业界的广泛关注^[1]. 路径规划是无人驾驶领域的核心技术之一. 车辆的路径规划指的是已知车辆的起始状态、目标状态和环境中的障碍物分布, 规划出一条与障碍物不相碰撞的且满足车辆的

运动学约束、环境约束和时间约束的可行路径^[2].

路径规划算法在移动机器人领域得到了广泛研究, 许多经典的路径规划算法被提出, 主要有人工势场法、可视图法、数学规划法等. 人工势场法实时性好, 便于实现, 但是容易陷入局部极小值且不利于增加约束条件^[3]. 可视图法能求得最短路径, 但搜索时间长, 并

① 基金项目: 国家自然科学基金(61503362, 91420104); 安徽省自然科学基金(1508085MF133)

收稿时间: 2016-08-17; 采用时间: 2016-09-08

缺乏灵活性^[4]. 数学规划法便于考虑机器人自身的约束条件, 但是计算量比较大^[5].

RRT 算法^[6,7]作为一种基于采样的增量式快速搜索算法, 可以弥补上述算法存在的路径搜索速度慢和不易于考虑约束条件的缺陷. 但是由于基本 RRT 算法采用均匀采样的策略, 缺乏引导信息, 导致算法规划出的路径迂回曲折, 在复杂的障碍物分布环境下, 算法的执行时间比较长^[8].

针对上述不足, 各种改进的 RRT 算法被提出. 为了解决 RRT 算法的最优性问题, Karaman 和 Frazzoli 提出了 RRT* 算法^[9], 在搜索树扩展的过程中树的结构并不是固定不变的, 而是根据最优准则来调整树中节点的父节点, 保证了算法的近似最优性, 但是该算法需要充足的时间来寻找最优路径. 为了解决生成路径迂回曲折的问题, Ma 等人针对城市道路环境提出了一种快速的 RRT 算法^[10], 根据不同的驾驶行为如直行、左转和右转等, 使用预先生成的模板初始化树, 从而得到引导信息, 提高了搜索效率, 解决了生成路径的迂回曲折问题, 但是该算法只适合城市道路环境下的路径规划, 不能解决非结构化环境中的路径规划问题. 为了解决复杂环境下规划时间长的问题, LaValle 和 Kuffner 提出了 RRT-connect 算法^[11,12], 来提高节点扩展的效率. 但是 RRT-connect 算法规划出的路径仍存在迂回曲折的缺点.

为了能够解决路径规划中的最短路径问题, Hart^[13]等人提出了 A* 算法, A* 算法是一种基于栅格地图的启发式搜索算法, 具有最优性和完备性的特点. 但是, 由于 A* 算法以栅格的中心点作为状态空间里的可搜索节点, 其可搜索邻域的个数为 8 个或 4 个, 因此路径的搜索方向被限制为 $\pi/4$ 的整数倍, 所以 A* 算法规划出的路径转折太多, 不够平滑, 一般不能被智能车辆直接使用^[14].

本文提出了一种基于 A* 引导域的 RRT 算法 (Guiding-Area RRT algorithm based on A*), 下文中简记为 GA-RRT 算法, 用来解决智能车辆的路径规划问题. 该算法结合了 A* 算法的最优性、完备性和 RRT 算法的快速性、便于考虑运动学约束的特性. 首先, 对环境地图进行低分辨率的栅格化处理, 然后利用 A* 算法在低分辨率的栅格地图上进行路径规划, 实时地规划出一条最短路径. 然后根据 A* 算法规划的结果生成引导域. 在 RRT 采样的过程中, 将目标偏向采样策略^[15,16]和

引导域偏向采样策略相结合. 在 RRT 算法的最近邻搜索函数中加入对车辆自身约束的考虑. 最后, 对最终生成的规划路径利用三次 B 样条进行曲线拟合, 以保证生成路径的平滑性. 通过与基本 RRT 算法和 RRT-connect 算法的仿真对比实验和实车对比测试, 验证了 GA-RRT 算法的优越性、有效性和实用性.

2 A* 算法

2.1 传统 A* 算法

A* 算法是基于栅格地图的启发式搜索算法, 其寻找从起始栅格节点 s_{start} 到目标栅格节点 s_{goal} 的最优路径: 将 s_{start} 添加到 OPEN 列表中, 接下来将与 s_{start} 相邻的栅格节点 $s_{near} \in S_{free}$ 添加到 OPEN 列表中, S_{free} 为无障碍栅格空间, 并将 s_{near} 的父节点设为 s_{start} . 然后将 s_{start} 从 OPEN 列表中删除并加入到 CLOSE 列表中, OPEN 列表中下一个被处理的节点由 $f=g+h$ 决定, g 为从 s_{start} 到当前栅格节点的实际路径代价, h 为从当前栅格节点到 s_{goal} 的估计路径代价, 将具有最小 f 值的栅格节点 s_{best} 从 OPEN 列表中选出, 对其采取与 s_{start} 相似的处理过程. h 值必须小于从当前栅格节点到 s_{goal} 的实际最优路径代价值, 才能保证最终规划出的路径的最优性.

2.2 低分辨率环境地图下的 A* 算法

在 A* 算法的实际使用中, 通常先将环境地图进行栅格化处理, 将每一个栅格当作一个状态, 而当栅格的尺寸越大, 即地图的分辨率越低, 则环境地图分解成的状态数越少, 使得在当起始点坐标和目标点坐标相同时, A* 算法的执行时间会变短, 规划出的栅格路径也会发生相应变化. 图 1 环境的大小为 100×100 , 以地图的左下角为坐标系原点, 以水平向右的方向为 x 轴的正方向, 以垂直向下的方向为 y 轴的正方向. 起点和终点的坐标分别为 $(2, 100)$ 和 $(100, 2)$. 针对图 1 的障碍物环境, 分别进行 512×512 、 256×256 、 128×128 和 64×64 的栅格化, 规划结果如图 1 中的 (a)、(b)、(c) 和 (d) 所示.

四种不同分辨率环境地图下 A* 算法的规划时间如图 2 所示, 从图中可以看出, A* 算法的规划时间与栅格地图分辨率的平方成正比例关系, 分辨率的降低可以大幅减少规划时间, 同时在不同分辨率的栅格地图上, A* 算法规划出的路径的趋势是一致的.

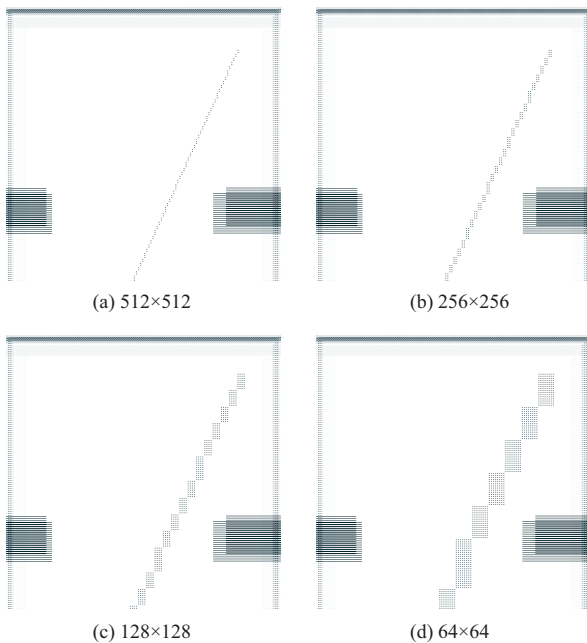


图1 不同分辨率栅格地图下 A*算法规划结果

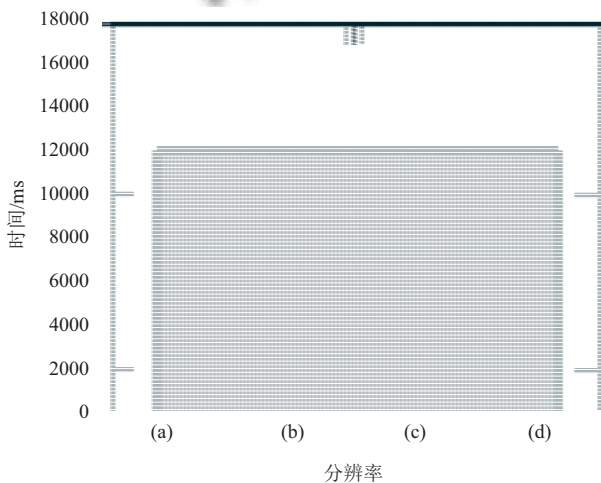


图2 不同分辨率栅格地图下 A* 算法规划时间

3 基于 A* 引导域的 RRT 算法

3.1 基本 RRT 算法

本文研究二维空间里智能车辆的路径规划问题, 定义智能车辆的状态空间为 $X = R^2$, 在欧氏空间中障碍物占据的空间为 $X_{ob} \subset X$, 无障碍空间为 $X_{free} = X \setminus X_{ob}$, 路径规划问题就是求解:

$$\begin{cases} x(t) \in X_{free}, t \in [0, t_f] \\ x(0) = x_{start} \\ x(t_f) = x_{goal} \end{cases} \quad (1)$$

x_{start} 为机器人的起始状态, x_{goal} 为目标状态. 基本的 RRT 算法的思想是以起始状态 x_{start} 来初始化树 T ,

然后以迭代的方式在无障碍空间 X_{free} 中随机采样状态 x_{rand} , 然后遍历树 T 中的节点, 搜索距离 x_{rand} 最近的状态 $x_{nearest}$, 机器人由 $x_{nearest}$ 状态向 x_{rand} 状态直线行驶产生新的状态 x_{new} , 若在行驶的过程中与障碍物不相碰撞, 则将 x_{new} 状态和从 $x_{nearest}$ 到 x_{new} 的路径添加到树 T 中, 重复上述过程, 直到目标状态 x_{goal} 添加到树 T 中为止. 最后由目标状态 x_{goal} 回溯到起始状态 x_{start} 得到规划路径. 其搜索过程如 Algorithm 1 所示. Algorithm 1 第 2 行中的 $notInTree(x_{goal})$ 函数用来判断目标状态 x_{goal} 是否在树 T 中, 如在树 T 中则返回 false, 否则返回 true. 第 7 行和第 8 行的作用是将满足条件的节点和边添加到树中. 第 11 行返回基本 RRT 算法的规划路径.

算法1: 基本RRT算法

```

1.  $T.init(x_{start})$ 
2. while( $T.notInTree(x_{goal})$ )
3.    $x_{rand} = Random()$ 
4.    $x_{nearest} = T.Nearest(x_{rand})$ 
5.    $x_{new} = Steer(x_{nearest}, x_{rand})$ 
6.   if( $\neg Collision(x_{nearest}, x_{new})$ )
7.      $T.addNode(x_{new})$ 
8.      $T.addEdge(x_{nearest}, x_{new})$ 
9.   end if
10. end while
11.  $path = T.getPathFromTree()$ 
12. return path

```

3.2 GA-RRT 算法

为了能够实时地规划出智能车辆可执行的路径, 本文在基本 RRT 算法的框架下提出了 GA-RRT 算法, 该算法的主要思想是先在低分辨率的栅格图上用 A* 算法进行路径规划, 然后根据 A* 算法的规划结果生成引导域 X_{guide} , 采用目标偏向采样和引导域偏向采样相结合的策略进行随机采样, 并在最近邻搜索过程中增加角度约束. 最后, 对生成的折线段路径利用三次 B 样条进行曲线拟合^[17]. 完整的算法见 Algorithm 2.

算法2: GA-RRT算法

```

1.  $X_{guide} = generateGuideSpace()$ 
2.  $T.init(x_{start})$ 
3. while( $T.notInTree(x_{goal})$ )

```

```

4.  $x_{rand} = nonUniformRandom()$ 
5.  $x_{nearest} = T.limitNearest(x_{rand})$ 
6.  $x_{new} = Steer(x_{nearest}, x_{rand})$ 
7.  $if(!Collision(x_{nearest}, x_{new}))$ 
8.    $T.addNode(x_{new})$ 
9.    $T.addEdge(x_{nearest}, x_{new})$ 
10.  $end\ if$ 
11.  $end\ while$ 
12.  $path = T.getPathFromTree()$ 
13.  $post\_path = BsplineFitting(path)$ 
14.  $return\ post\_path$ 

```

3.3 A*引导域

记 A* 算法规划路径经过的第 i 个栅格中心点坐标为 x_i , 则引导域的生成公式如(2)和(3)所示:

$$X_i = \{x | \|x - x_i\| \leq r\} \quad (2)$$

$$X_{guide} = \bigcup_{i=1}^n X_i \quad (3)$$

r 为单个路径点生成的引导域的半径大小。图 3 为引导域示意图, 绿色的圆点为 A* 算法规划路径上的栅格的中心点, 红色圆圈内的区域为生成的引导域。

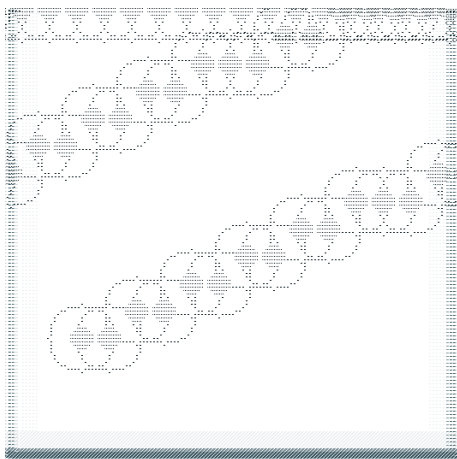


图 3 引导域

3.4 采样策略

基本 RRT 算法对整个无障碍空间进行均值采样, 这样虽然增加了搜索到可行路径的可能性, 但是采样区域过于分散导致很多不必要的采样, 减缓了算法的收敛速度。

在 3.3 节引导域的基础上, 本文提出了目标偏向采

样和引导域偏向采样相结合的采样策略, 既保证了采样的随机性, 又增加了算法的收敛速度。详细步骤如函数 $nonUniformRandom()$ 所示。

函数: $nonUniformRandom()$

```

1.  $p = rand(0, 1)$ 
2.  $if(p < p_{goal\_bias})$ 
3.    $x_{rand} = x_{goal}$ 
4.  $else$ 
5.    $p = rand(0, 1)$ 
6.    $if(p < p_{choose})$ 
7.      $x_{rand} = uniRandom(X_{guide})$ 
8.    $else$ 
9.      $x_{rand} = uniRandom(X_{free})$ 
10.   $end\ if$ 
11.  $end\ if$ 
12.  $return\ x_{rand}$ 

```

3.5 最近邻搜索

最近邻搜索的功能是寻找树中距离随机采样节点最近的节点, 因为智能车辆在行驶的过程中有最大转向角度限制, 所以本文在寻找树 T 中距离 x_{rand} 最近的节点时, 加入角度约束, 使得从当前节点 x 的父节点 x_{par} 指向节点 x 的向量 $\overrightarrow{x_{par}, x}$ 与从当前节点 x 指向 x_{rand} 的向量 $\overrightarrow{x, x_{rand}}$ 之间的夹角小于车辆的最大转向角。具体步骤如函数 $limitNearest(x_{rand})$ 所示。

函数: $limitNearest(x_{rand})$

```

1.  $minDis = \infty, x_{nearest} = x_{start}$ 
2.  $for\ every\ x\ in\ T$ 
3.    $if\ Ang(\overrightarrow{(x_{par}, x)}, \overrightarrow{(x, x_{rand})}) < \alpha_{min}$ 
4.      $if\ dis(x, x_{rand}) < minD$ 
5.        $minD = dis(x, x_{rand})$ 
6.        $x_{nearest} = x$ 
7.      $end\ if$ 
8.    $end\ if$ 
9.  $end\ for$ 
10.  $return\ x_{nearest}$ 

```

4 实验与分析

本文分别进行了仿真实验和实车测试。仿真实验

环境为狭窄通道环境,通过对比基本 RRT 算法、RRT-connect 算法和 GA-RRT 算法的规划性能,验证 GA-RRT 算法的优越性和有效性.利用无人车平台“智能先锋 II”,如图 4 所示,分别在非结构化的停车场环境和结构化的弯道环境里进行实车测试,验证了 GA-RRT 算法的实用性.

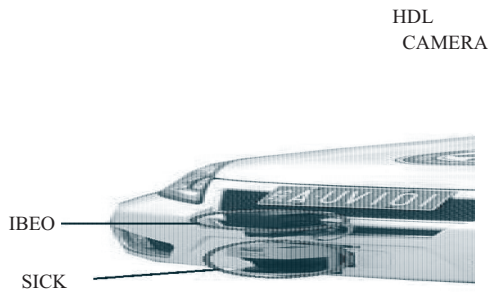


图 4 智能车“智能先锋 II”

4.1 仿真实验

仿真实验程序运行在 win7 操作系统上,PC 机的处理器为 Core i5,内存为 4 GB.图 5 的仿真环境大小为 500×500 ,车辆以矩形框表示,蓝色的矩形框表示车辆的起点,绿色的矩形框表示车辆的终点,蓝色和红色线条分别表示用三次 B 样条平滑前后的路径,绿色圆点表示扩展树中的节点,灰色的线段表示扩展树中的边.在图 5 中,(a)和(b)分别表示用基本 RRT 算法规划出的路径平滑前后的结果;(c)和(d)分别表示用 RRT-connect 算法规划出的路径平滑前后的结果;(e)和(f)表示用 GA-RRT 算法规划出的路径平滑前后的结果,在生成引导域时,环境地图进行了 50×50 的栅格化.针对图 5 的仿真实验环境,分别用三种算法做了 20 次测试,实验结果见表 5.

从图 5 和表 1 中可以看出,GA-RRT 算法的规划时间最短,且它规划出的路径的最大曲率值最小,表明其搜索效率最高且规划出的路径最平滑.同时,GA-RRT 算法规划出的路径的平均长度优于基本 RRT 算法和 RRT-connect 算法的规划结果,使得路径长度的最优性得到改善.

4.2 实车测试

“智能先锋 II”的可行驶路径的最大曲率为 0.16 m^{-1} ,最大转向角为 30° ,采用三维激光雷达 Velodyne 感知周围的环境,并以 120 ms 为周期实时构建 512×512 的栅格地图,栅格地图中栅格的宽度为 0.2 m .

图 6 为非结构化的停车场环境下的实验结果.(a)为停车场场景图,(b)、(c)和(d)中白色区域为可通行区域,黑色区域为障碍物区域,红色线条表示规划结果,蓝色矩形框和绿色矩形框分别为车辆的起始位置和目標位置,绿色圆点表示扩展树中的节点,深灰色线段表示扩展树中的边.(b)为基本 RRT 算法的规划路径经过三次 B 样条拟合后的结果;(c)为 RRT-connect 算法的规划路径经过三次 B 样条拟合后的结果;(d)为 GA-RRT 算法的规划结果,此时栅格地图的分辨率为 64×64 .

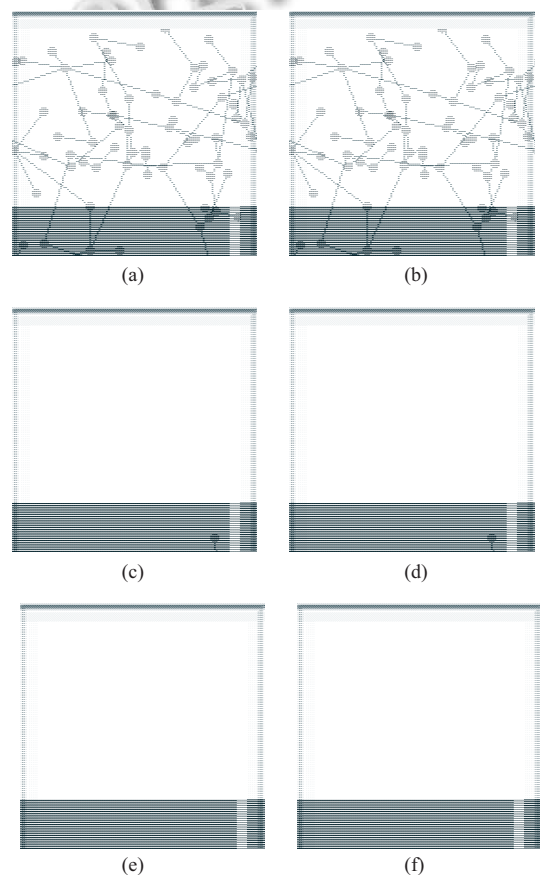


图 5 3 种算法路径规划平滑前后结果

表 1 仿真实验数据对比

20次实验	平均时间 (ms)	平均路径长度 (m)	路径最大曲率 (m^{-1})	平均采样 次数
Basic-RRT	203.57	189.42	0.83	363
RRT-connect	75.21	169.42	0.82	127
GA-RRT	72.17	141.32	0.15	110

在停车场测试环境下,对三种算法进行了 20 次统计试验,相关数据记录在表 6 中.

从图6和表2中可以看出 GA-RRT 算法的规划时间远低于基本 RRT 算法, 提高了搜索效率, 虽然其规划时间略高于 RRT-connect 算法, 但是改善了生成路径的长度和生成路径的最大曲率. 由于 GA-RRT 算法的规划周期小于地图更新周期 120 ms 且路径的最大曲率小于智能车可行驶路径的最大曲率 0.16 m^{-1} , 因此满足智能车辆的实时运动规划需求.

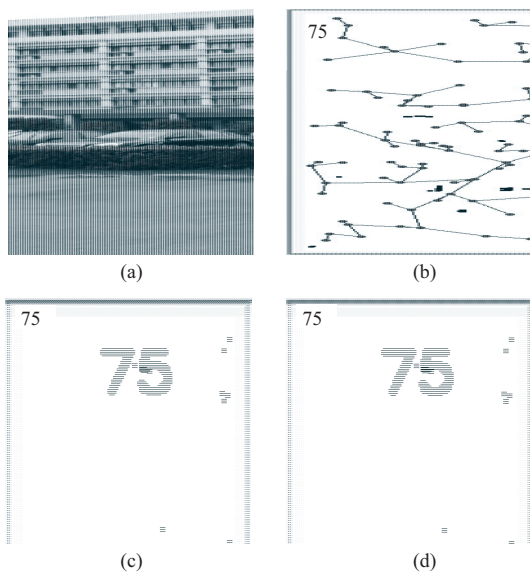


图6 停车场实车实验

表2 停车场实验数据对比

20次实验	平均时间 (ms)	平均路径长度 (m)	路径最大曲率 (m^{-1})	平均采样次数
Basic-RRT	270.00	53.43	0.85	317
RRT-connect	10.57	35.59	0.81	15
GA-RRT	30.57	29.43	0.11	32

图7为结构化的弯道场景实验结果, 其中的(a)弯道的场景图. (b)、(c)和(d)表示的含义与图6中的(b)、(c)和(d)相同. 在使用 GA-RRT 算法规划时, 环境地图进行 64×64 的栅格化处理. 从图7和表7中可知, 在三种算法中, GA-RRT 算法规划时间远优于基本 RRT 算法, 提高了搜索效率, 虽然规划时间略高于 RRT-connect 算法, 但是其规划路径的长度最短、路径的最大曲率值最小. 因为其规划周期小于地图更新周期, 路径的最大曲率低于智能车辆可行驶路径的最大曲率, 所以能够满足智能车辆的实时运动规划需求.

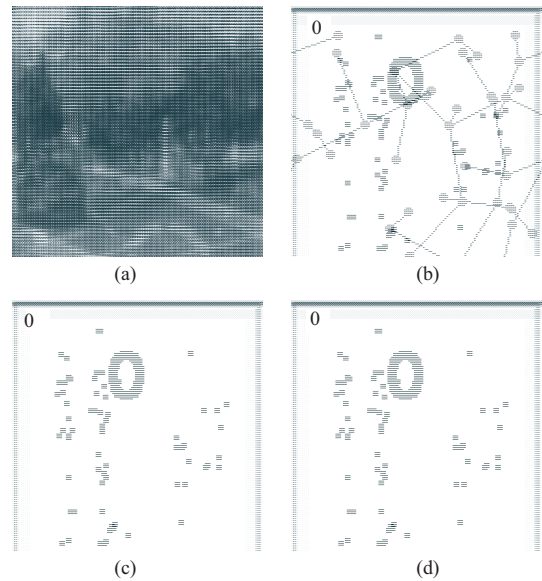


图7 弯道换道实车实验

表3 弯道换道实验数据对比

20次实验	平均时间 (ms)	平均路径长度 (m)	路径最大曲率 (m^{-1})	平均采样次数
Basic-RRT	150.00	44.89	0.89	294
RRT-connect	4.21	38.29	0.83	10
GA-RRT	9.37	29.16	0.10	21

5 结语

鉴于基本 RRT 算法由于存在采样效率低且缺少对车辆自身约束的考虑等不足, 使得其难以满足智能车辆这类轮式移动机器人的应用需求. 对此, 本文提出了基于 A*引导域的 RRT 算法, 使用目标偏向与 A*引导域偏向相结合的采样策略, 以提升采样效率. 此外, 在最近邻搜索函数的设计中, 增加了对车辆运动学约束的考虑, 并用三次 B 样条对生成的路径进行平滑处理, 以使得最终生成的路径能够被车辆执行. 通过仿真实验和实车实验验证了该算法在规划时间、路径长度和最大曲率三个方面的优越性, 证实了其在智能车辆上的有效性和实用性.

参考文献

- 1 Chu K, Lee M, Sunwoo M. Local path planning for off-road autonomous driving with avoidance of static obstacles. IEEE Trans. on Intelligent Transportation Systems, 2012, 13(4): 1599-1616. [doi: 10.1109/TITS.2012.2198214]
- 2 刘华军, 杨静宇, 陆建峰, 等. 移动机器人运动规划研究综述. 中国工程科学, 2006, 8(1): 85-94.

- 3 Raja P, Pugazhenti S. Optimal path planning of mobile robots: A review. *International Journal of Physical Sciences*, 2012, 7(9): 1314–1320.
- 4 陈少斌. 自主移动机器人路径规划及轨迹跟踪的研究[博士学位论文]. 杭州: 浙江大学, 2008.
- 5 Masehian E, Sedighizadeh D. Classic and heuristic approaches in robot motion planning—a chronological review. *World Academy of Science, Engineering and Technology, International Journal of Mechanical, Aerospace, Industrial, Mechatronic and Manufacturing Engineering*, 2007, 1(5): 228–233.
- 6 LaValle SM, Kuffner JJ. Randomized kinodynamic planning. *The International Journal of Robotics Research*, 2001, 20(5): 378–400. [doi: [10.1177/02783640122067453](https://doi.org/10.1177/02783640122067453)]
- 7 LaValle SM. From dynamic programming to RRTs: Algorithmic design of feasible trajectories. Bicchi A, Prattichizzo D, Christensen HI. *Control Problems in Robotics*. Berlin Heidelberg, Germany: Springer, 2003. 19–37.
- 8 Elbanhawi M, Simic M. Sampling-based robot motion planning: A review. *IEEE Access*, 2014, 2: 56–77. [doi: [10.1109/ACCESS.2014.2302442](https://doi.org/10.1109/ACCESS.2014.2302442)]
- 9 Karaman S, Frazzoli E. Sampling-based algorithms for optimal motion planning. *The International Journal of Robotics Research*, 2011, 30(7): 846–894. [doi: [10.1177/0278364911406761](https://doi.org/10.1177/0278364911406761)]
- 10 Ma L, Xue JR, Kawabata K, *et al.* A fast RRT algorithm for motion planning of autonomous road vehicles. 2014 IEEE 17th International Conference on Intelligent Transportation Systems. Qingdao, China. 2014. 1033–1038.
- 11 LaValle SM, Kuffner JJ Jr. Rapidly-exploring random trees: Progress and prospects. 4th International Workshop on Algorithmic Foundations of Robotics. Wellesley, USA. 2000. 293–308.
- 12 Kuffner JJ, LaValle SM. RRT-connect: An efficient approach to single-query path planning. *IEEE International Conference on Robotics and Automation*. San Francisco, USA. 2000. 995–1001.
- 13 Hart PE, Nilsson NJ, Raphael B. A formal basis for the heuristic determination of minimum cost paths. *IEEE Trans. on Systems Science and Cybernetics*, 1968, 4(2): 100–107. [doi: [10.1109/TSSC.1968.300136](https://doi.org/10.1109/TSSC.1968.300136)]
- 14 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. *机器人*, 2014, 36(5): 627–633.
- 15 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. *机器人*, 2015, 37(4): 443–450.
- 16 徐娜, 陈雄, 孔庆生, 等. 非完整约束下的机器人运动规划算法. *机器人*, 2011, 33(6): 666–672.
- 17 Berglund T, Brodnik A, Jonsson H, *et al.* Planning smooth and obstacle-avoiding B-spline paths for autonomous mining vehicles. *IEEE Trans. on Automation Science and Engineering*, 2010, 7(1): 167–172. [doi: [10.1109/TASE.2009.2015886](https://doi.org/10.1109/TASE.2009.2015886)]