

改进 Basic Theta* 中 LOS 算法的任意角度路径规划

陈正泽¹, 彭力¹⁺, 戴菲菲²

1. 物联网技术应用教育部工程研究中心 (江南大学 物联网工程学院), 江苏 无锡 214122

2. 台州市产品质量安全检测研究院, 浙江 台州 318000

+ 通信作者 E-mail:jnpengli@outlook.com

摘要: 针对传统 A* 算法在进行机器人路径规划时, 生成的路径只能沿网格方向进行, 带来冗余路径的产生, 导致规划的路径非最优路径等问题, 提出了一种对 A* 算法的改进策略。首先, 设计 Basic Theta* 算法, 改进 A* 算法的搜索方式, 通过引入视线算法(line of sight, LOS), 增加搜索过程中非网格方向路径的选择, 实现任意角度的路径规划; 其次, 对 Basic Theta* 算法中的 LOS 进行改进, 提出一种基于网格中心点的视线检测算法, 提高 Basic Theta* 算法规划路径的可实现性。为了验证改进 A* 算法和改进 Basic Theta* 算法的效果, 在五种不同尺寸的栅格地图中进行仿真测试, 仿真结果表明, 改进后的 A* 算法能够在运行时间基本不变的情况下实现路径沿任意角度进行, 打破网格的限制, 剔除冗余路径, 降低路径长度, 路径长度的缩减率可达到 8%; 改进的 Basic Theta* 算法在保证路径长度基本不变的情况下, 能够有效优化传统 Basic Theta* 算法路径中存在的不符合机器人运动学的路径, 同时缩短 Basic Theta* 算法的运行时间, 提高算法的运行效率, 规划的路径能够满足机器人运动的实际要求。

关键词: 路径规划; Basic Theta* 算法; 任意角度路径; 改进 LOS

文献标志码: A **中图分类号:** TP242

Arbitrary angle path planning based on improved LOS algorithm in Basic Theta*

CHEN Zhengze¹, PENG Li¹⁺, DAI Feifei²

1. Engineering Research Center of Internet of Things Technology Applications, School of IoT Engineering, Jiangnan University, Wuxi, Jiangsu 214122, China

2. Taizhou Institute of Product quality and Safety testing, Taizhou, Zhejiang 318000, China

Abstract: When the traditional A* algorithm is used for robot path planning, the generated path can only be carried out along the grid direction, which leads to the generation of redundant paths and leads to the non optimal path of the planned path. So an improved strategy for the A* algorithm is proposed. Firstly, the Basic Theta* algorithm is designed to improve the search mode of A* algorithm. By using the line of sight algorithm, the selection of non grid path in the search process is increased to realize path planning from any angle; Secondly, the line of sight algorithm in Basic Theta* algorithm is improved, and a line of sight detection algorithm based on grid center point is proposed

* National Key R&D Program of China No.2018YFD0400902 (国家重点研发计划); The National Natural Science Foundation of China under Grant No. 61873112 (国家自然科学基金项目).

to improve the realizability of Basic Theta* algorithm. In order to verify the effect of the improved A* algorithm and the improved Basic Theta* algorithm, the simulation test is carried out in five grid maps of different sizes. The simulation results show that the improved A* algorithm can realize the path along any angle under the condition of basically unchanged running time, break the limitation of the grid, eliminate redundant paths, reduce the path length, and increase with the increase of map size, the more obvious the effect of path shortening; At the same time, the improved Basic Theta* algorithm can effectively optimize the path that does not conform to the robot kinematics in the path of the traditional Basic Theta* algorithm, shorten the running time of the Basic Theta* algorithm, improve the operation of the algorithm, and the planned path can meet the actual requirements of robot motion.

Key words: path planning; Basic Theta* algorithm; any angle path; improved LOS

路径规划技术是机器人研究领域中的一个重要分支, 所谓机器人的最优路径规划问题, 就是依据某个或者某些最优原则, 例如工作代价最小、行走路线最短或者行走时间最短等, 在规定区域的范围内, 寻找一条从起始状态到目标状态的最优或者近似最优的安全路径^[1-4]。路径规划算法包括深度优先搜索、广度优先搜索、贪婪最佳优先算法、Dijkstra、A*等各种算法^[5-8]。其中A*算法在路径规划中具有广泛的应用, 通过在评估函数中引入启发函数, 兼顾算法效率和完备性。A*算法可以很快地寻找出一条附着在网格边缘上的路径, 但是由于其算法中路径点沿网格点排布的约束, 导致A*算法在寻路过程中可形成的角度往往是被网格形状固定的, 在寻路的过程中, 这种路径看起来十分不自然。并且A*算法寻找出来的路径往往并不是实际地形下真正最短路径, 而规划出任意航向的最短路径恰恰是机器人移动的迫切要求。

为了解决这一问题, 在A*算法基础上, 研究人员提出了一些改进算法。A* PS(A* with post-smoothing)算法^[9-11]对A*寻找的路径进行后续处理, 得到平滑的路径, A* PS同样也是基于网格的, 但是在其寻找到路径之后, 通过一定的后处理算法, 对其找到的路径进行平滑处理, A* PS通常可以产生比A*更短的路径, 但是它并不能保证获取到真实情况下的最短路径, 同时也会增加整个寻路过程的耗时。Ferguson D等人提出的Field D*算法^[12-14]采用线性插值, 使得路径点不局限于栅格中心

或四角, 消除了相邻栅格之间的离散状态转移所带来的路径不平滑问题, 但是路径中经常存在不必要的转向和计算量较大的缺点。高博^[15]等人针对Field D*该方法的计算量很大的缺点, 尤其是在对动态目标的规划时, 计算时间无法满足实际要求, 在对Field D*计算原理分析的基础上, 提出了在目标是动态的情况下, 只进行前目标与新目标的路径规划, 并对Field D*计算所得路径消耗只进行局部修改的动态规划算法, 该动态算法可以有效减少计算量。Nash A等人结合A*算法和可视图法的优点, 提出了Basic Theta*和Angle Propagation Theta*(统称为Theta*)算法^[16-17], 通过引入可视性检测来寻找捷径, 找到任意方向的路径, 但是在规划路径的正确性和合理性方面存在缺点。王琼等人提出了一种基于地形危险度的任意航向路径规划算法Risk Theta*^[18], 以地形特征统计分析为基础提出了地形危险度指标, 并建立地形危险度地图, 在此基础上应用Basic Theta*搜索, 以危险度最低为方向搜索最优路径, 该算法能够在栅格地图上找到比A*算法危险度低得多任意航向路径, 具有较强的实用性。肖国宝等人^[19]对Theta*算法中的启发函数进行修改, 将障碍物对机器人产生的斥力作为一种惩罚函数加入启发函数中, 并通过试探证明了算法的合理性和有效性。毛杰等人^[20]采用Lazy Theta*算法对反潜巡逻飞机的飞行航线进行规划, 并将这一过程进行计算机实现, 为反潜人员决策提供有力支撑。FARIAM等人^[21]在稀疏网格中将Lazy Theta*算法应用到多

旋翼无人机上，用于搜索大型的3D空间。DERGACHEV S等人^[22]将Theta*、ORCR和Push and Rotate融合之后用于多智能体的导航。

目前对Theta*算法的改进主要集中在改进启发函数、改进视线检测算法的时机和融合其他辅助算法等，对于视线检测算法的改进尚且没有涉及，因此本文在Basic Theta*算法实现任意角度路径规划的基础上，做了如下改进：

(1)针对Basic Theta*算法中使用的传统视线检测算法存在检测误差的缺点，本文改进视线检测算法的搜索原理，提出一种完全保证可视性检测正确性的LOS算法。

(2)针对Basic Theta*使用传统视线检测算法规划的路径存在不合理路径，本文提出将网格中心点替代网格顶点作为检测节点，实现优化路径的不合理部分；

1 问题描述

1.1 环境建模

全局路径规划是基于环境模型的规划方法，移动机器人在规划之前通过传感器获得环境的二维模型。单元分解建模是典型的建模方法，其主要思想是将环境离散化为若干个规则的相同大小的基本单元，通过坐标信息便可以对障碍物和非障碍物进行标识。

本文使用经典的栅格法来完成环境建模。如图1所示，将移动机器人的工作环境切割成相同大小且互相连接的栅格，不同的坐标信息表示不同位置的栅格。根据环境信息将栅格分成了两种类型：障碍空间和自由空间。图中白色栅格表示移动机器人可以通过的区域，属于自由空间；黑色栅格表示环境中的障碍物，属于障碍空间，此时机器人不能通过。

为了便于研究和开展实验，在栅格建模过程中将障碍物进行膨胀宽度为机器人半径的膨胀处理，形成构型空间，同时假设在转角处互为对角的栅格可以通过。因此可以将机器人看作质点。

A*算法中，在不存在边界和障碍物的情况下，如图2所示，机器人可以到达相邻栅格的各个顶点。

机器人可运动方向只有8个。

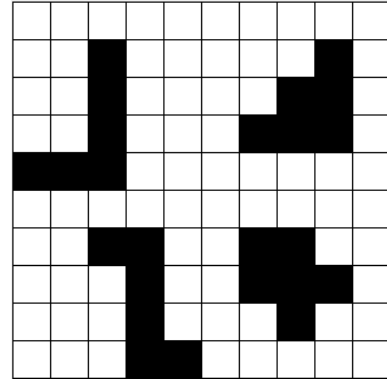


Fig.1 Grid Map

图1 栅格地图

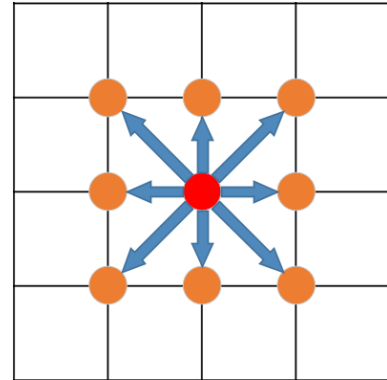


Fig.2 Motion direction of A* algorithm

图2 A*算法机器人运动方向

1.2 A*算法基本思想

A*算法是可实现全局路径规划的启发式搜索算法，它根据定义的估价函数在静态环境下搜索最优路径。A*算法的估价函数 $f(n)$ 表示为：

$$f(n) = g(n) + h(n) \quad (1)$$

在式(1)中， $f(n)$ 表示从起始点经由任意节点 n 到达目标点的估价函数， $g(n)$ 表示起始点到节点 n 的实际代价， $h(n)$ 表示节点 n 到目标点的估计代价。本文采用曼哈顿距离度量两点之间的估计代价：

$$d = |x_1 - x_2| + |y_1 - y_2| \quad (2)$$

在式(2)中， (x_1, y_1) 、 (x_2, y_2) 分别表示两节点 $n1$ 、 $n2$ 的坐标，通过比较 $f(n)$ ，选取代价最小的候选节点作为下一路径节点。

A*算法的实现步骤如下：

1) 以起点作为第1个路径节点, 寻找与之相连的候选节点;

2) 对与之相连的每一个候选节点, 计算其启发函数值, 选出函数值最小的节点作为下一个路径节点;

3) 如果该节点是目标点, 则终止规划, 否则, 进入第2步, 以新的路径节点为起点继续规划, 并记录搜索到的最短路径。

1.3 A*算法的缺点

A*算法在搜索过程中通过引入启发信息来实现在向目标节点移动的判别, 无需遍历地图, 能够快速高效地找出一条最优的路径。

但是, 作为基于网格的规划器, A*算法面临着严重的限制。由于其搜索的方向和路径的方向只能按照相邻八个栅格固定45度偏角来行进, 导致A*算法在寻路过程中可形成的角度往往是被网格形状固定的, 如图3(a)所示, 生成的路径会被人为地强制地附着在网格边上, 导致不必要的路径代价产生, 从而得到的路径为非最短路径。而真正的最短路径通常不遵循网格线且可以任意方向行进, 如图3(b)所示。

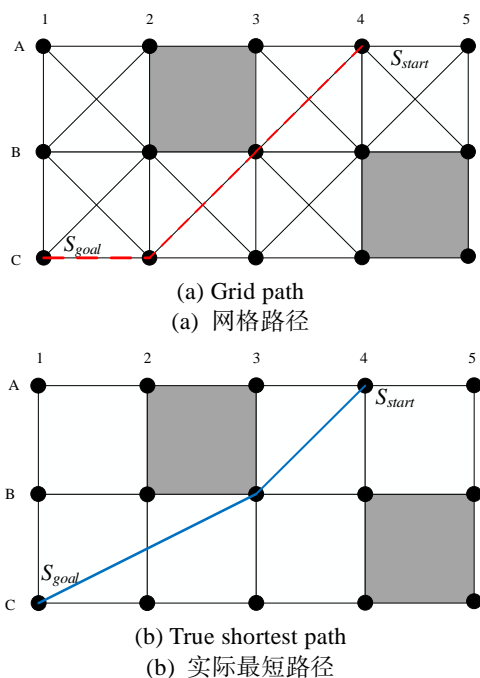


Fig.3 Grid path versus true shortest path

图3 网格路径和实际最短路径对比

2 改进 A*算法

本文为了解决A*算法路径规划过程中存在路线被限制在网格边, 路径非最短路径等问题, 首先在2.1节中设计Basic Theta*算法, 借助可视性检测算法对A*算法的搜索方式进行改进, 实现任意角度路径规划; 针对Basic Theta*算法使用的传统可视性检测算法存在检测误差的问题, 同时为了克服基于网格顶点检测的传统视线算法规划的路径存在不合理的部分, 在2.2节中提出一种基于网格中心点检测的改进可视性检测算法, 避免可视性检测误差, 同时克服不合理路径的产生。

2.1 Basic Theta*算法

Basic Theta*算法是一种栅格路径规划算法, 其最大的改进在于它不要求搜索树节点的邻节点必须是其相邻的节点, 因此所求解路径的轨迹不限制在栅格的横向、纵向和对角方向, 理论上允许以任意角度改变路径的方向。

Basic Theta*算法与A*算法的主要差别在扩展节点的过程中进行必要的可视性检测和基于可视性检测的结果进行交错搜索平滑。

2.1.1 可视性检测算法(Line-of-sight)

可视性检测的原理是如果在两个目标之间绘制的假想线不与任何障碍物相交, 则认为视线存在。可视性检测算法的原理是利用布雷森汉姆直线算法的基本规律。

布雷森汉姆直线算法是用来描绘由两点所决定的直线的算法, 它会算出一条线段在 n 维光栅上最接近的点, 每一个节点代表一个栅格位置, 就可以将二维栅格上两个点之间的线光栅化, 用一系列离散的正方形(像素)来近似表现这条直线。

因此, 可视性检测算法的执行需要借鉴布雷森汉姆直线算法的搜索方式, 在搜索过程中增加对符合要求节点所代表栅格的类型判断, 只有当搜索出来的节点代表的栅格都是自由空间时, 才存在视线; 只要有一个节点代表的栅格是障碍空间, 一定不存在视线。布雷森汉姆算法的搜索过程就是借助一定原则选择一条线段在 n 维光栅上最接近的点,

如图4所示，节点选择的原则是比较 $d1$ 、 $d2$ 的大小。

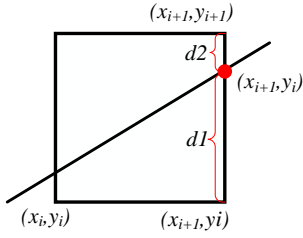


Fig.4 The selection principle of LOS

图4 可视性检测算法选择原则

Case1: 当 $d1 > d2$ 时，说明直线上理论点离 (x_{i+1}, y_{i+1}) 较近，下一点取 (x_{i+1}, y_{i+1}) 。

Case2: 当 $d1 < d2$ 时，说明直线上理论点离 (x_{i+1}, y_i) 较近，下一点取 (x_{i+1}, y_i) 。

假设目标直线方程为 $y = kx + b$ ，则在 x_{i+1} 处直线上的 y 坐标可计算为，因为 $x_{i+1} = x_i + 1$ ：

$$y = k(x_i + 1) + b \quad (3)$$

那么：

$$d1 = y - y_i = k(x_i + 1) + b - y_i \quad (4)$$

$$d2 = (y_i + 1) - y = y_i + 1 - k(x_i + 1) - b$$

要比较两段距离的大小，可以通过做差：

$$d1 - d2 = 2k(x_i + 1) - 2y_i + 2b - 1 \quad (5)$$

通过重新安排式(5)可获得算法第 k 步的决策函数 p_i ，从而仅使用整数计算。设 Δy 和 Δx 分别为两端点的垂直和水平偏移量， $k = \Delta y / \Delta x$ ，将决策参数定义为：

$$\begin{aligned} p_i &= \Delta x(d1 - d2) \\ &= 2\Delta y \cdot x_i - 2\Delta x \cdot y_i + b \end{aligned} \quad (6)$$

p_i 的符号与 $d1 - d2$ 的符号相同，以 Δx 大于零为例。参数 b 是一常量，其值为 $2\Delta y + \Delta x(2b - 1)$ ，它与像素位置无关，且会在循环计算 p_i 时被消除。假如 y_i 处的节点比 y_{i+1} 的节点更接近于线段(即 $d1 < d2$)，那么参数 p_i 是负的，此时，选择节点 (x_{i+1}, y_i) ；反之，选择节点 (x_{i+1}, y_{i+1}) 。

直线上的坐标会沿 x 或 y 方向的单位步长而变

化。因此，可以利用递增整数运算得到后继的决策参数值。在 $k+1$ 步，决策参数可以从式(6)计算得出：

$$p_{i+1} = 2\Delta y \cdot x_{i+1} - 2\Delta x \cdot y_{i+1} + b \quad (7)$$

将上述方程减去等式(3.14)，可以得到：

$$p_{i+1} - p_i = 2\Delta y(x_{i+1} - x_i) - 2\Delta x(y_{i+1} - y_i) \quad (8)$$

因为 $x_{i+1} = x_i + 1$ ，因而得到：

$$p_{i+1} = p_i + 2\Delta y - 2\Delta x(y_{i+1} - y_i) \quad (9)$$

其中， $y_{i+1} - y_i$ 取0或1，取决于参数 p_i 的符号，当 $p_i < 0$ ， $y_{i+1} - y_i$ 取0；当 $p_i > 0$ ， $y_{i+1} - y_i$ 取1。决策参数的递归运算在线段的坐标端点开始的每个整数 x 位置进行，起始节点位置 (x_0, y_0) 的第一个参数 p_0 通过式(6)及 $k = \Delta y / \Delta x$ 计算得出：

$$p_0 = 2\Delta y - \Delta x \quad (10)$$

通过对决策函数的递归计算，可以将接近目标直线的所有节点选择出来，对节点代表的栅格进行障碍物判断，当所有节点代表的栅格都不是障碍物，表示目标直线的起点和终点之间存在视线。传统的实现检测算法是通过近似的手段逼近目标直线，用离散的栅格序列近似的代替目标直线的形状，并不是目标直线经过的所有准确栅格，因此视线检测结果存在误差，这也是本文所优化的部分。

2.1.2 交错搜索和平滑

交错搜索和平滑就是将当前节点的相邻节点与当前节点的父节点做一次可视性检查，若可视，则比较从当前节点的父节点直接到当前节点的相邻节点，和从当前节点的父节点经当前节点到当前节点的相邻节点这两条路径的代价，选择代价更小的路径。

本文用 s 表示当前节点， s' 表示当前节点的相邻节点， s_{parent} 表示当前节点的父节点，用 $g(s)$ 表示 s 节点的实际代价，用 $c(s, s')$ 表示顶点 s 到顶点 s' 间的直线距离， $parent(s)$ 表示 s 节点的父节点，父节点就是 s 节点在路径中的上一步节点。

Basic Theta*交错搜索的实现很简单，当扩展一个新的顶点时，A*只会考虑一条路径，而Basic

Theta*会考虑两条路径,如图5所示。图5中,我们要从B3点(s)扩展到C3点(s'),其中,B3点的父节点是A4点(s_{start}),此时,Basic Theta*会考虑如下两条路径:

路线1: $s_{start} \rightarrow s \rightarrow s'$,此时需要将 s' 的实际代价和父节点更新为:

$$\begin{aligned} g(s') &= g(s) + c(s, s') \\ \text{parent}(s') &= s \end{aligned} \quad (11)$$

这也是A*考虑的路径,图5中用红色虚线表示。

路径2:如图5所示, $s_{start} \rightarrow s'$,此时需要将 s' 的实际代价和父节点更新为:

$$\begin{aligned} g(s') &= g(s_{start}) + c(s_{start}, s') \\ &= g(\text{parent}(s)) + c(\text{parent}(s), s') \\ \text{parent}(s') &= \text{parent}(s) \end{aligned} \quad (12)$$

这是Basic Theta*比A*额外考虑的路径,图5中用蓝色实线表示。

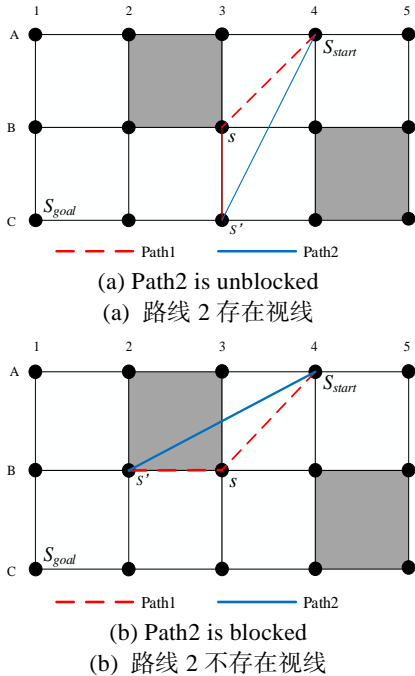


Fig.5 Path 1 and 2 considered by Basic Theta*

图 5 Basic Theta*搜索可选的两条路线

图5中 s 的父节点是起点 s_{start} ,情况比较特殊,一般情况下两条路线分别为:

路线1:第一段是由从起点 s_{start} 到 s 点的父节

点 s_{parent} 的路线;第二段是从 s 点父节点 s_{parent} 到 s 点的路线,记为 $lineA$;第三段是从 s 点到 s 点的邻接点 s' 的路线,记为 $lineB$ 。

路线2由两段组成:第一段是和路线1相同从起点 s_{start} 到 s 点的父节点的路线;第二段路线是从 s 的父节点到 s 邻接点 s' 的路线,记为 $lineC$ 。

$lineA$, $lineB$, $lineC$ 组成了一个三角形, $lineC$ 为三角形的斜边,根据三角形的边长性质,两边和一定大于第三边,可知路径2的距离一定比路径1更短。

交替搜索完成后通过对两条路线的选择来实现路线的平滑,选择路线的原则是基于可视性检测算法。首先路线2的长度是一定比路线1短,其次路线1是能够保证没有障碍的,但是路线2无法保证。因此对路线2中第二段 s 的父节点 s_{parent} 和 s 邻接点 s' 进行可视性检测,如果存在视线,如图5(a)所示,则Basic Theta*算法选择较短长度的路线2;如果不存在视线,如图5(b)所示,选择可以通行的路线1。

可以看出,从 $s_{start} \rightarrow s'$ 的路径,最短的路径即从 $s_{start} \rightarrow s'$ 的直线距离,但是按照A*算法的规划流程需要经过 $s_{start} \rightarrow s \rightarrow s'$,增加了冗余路径的产生。在引入可视性检测算法之后将中间节点 s 不予考虑,通过可视性检测算法来计算 $s_{start} \rightarrow s'$ 直线距离的可行性,从而到达最短路径的实现。当下一步拓展 s' 的子节点时,从 s_{start} 到 s' 子节点的路径必然存在中间节点 s' ,再通过可视性检测算法计算 s_{start} 到 s' 的子节点之间的直线距离是否可行,如果可行,路径将变成从 s_{start} 到 s' 的子节点的直线最短距离,进一步避免了冗余路径的产生。因此,增加可视性检测算法之后,可以在搜索过程中逐步减少路径的产生,最后通过积累,实现最终路径的削减。

在估价函数的选取上,Basic Theta*算法与A*算法是一致的,故在使用其解决路径规划问题时,式(1)作为其启发函数能够获取更优的路径。

图6显示了一个Basic Theta*寻路示例。每个顶

点标有各自节点到起始节点的代价和父节点，箭头指向顶点的父节点方向，被红色圈中的节点表示当前正在被扩展，蓝色箭头指出的顶点表示当前扩展过程中被搜索的节点。

第一步，像A*算法一样，Basic Theta*从起始节点A4(父节点为A4)开始扩展，将A4节点的邻接点的父节点更新为A4，如图6(a)所示。

第二步，从B3节点(父节点是A4)继续扩展，其中B2节点由于和B3节点的父节点A4之间不存在视线，因此Basic Theta*按照路线1来进行更新，将他的父节点设置为B3；节点C2、C3、C4和B3节点的父节点A4之间存在视线，因此Basic Theta*按照路线2来进行更

新，将他的父节点设置为A4，如图6(b)所示。

第三步，从B2节点(父节点是B3)继续扩展，其中A1、A2节点由于和B2节点的父节点B3之间不存在视线，因此Basic Theta*按照路线1来进行更新，将他的父节点设置为B2；节点B1、C1和B2节点的父节点B3之间存在视线，因此Basic Theta*按照路线2来进行更新，将他的父节点设置为B3，如图6(c)所示。

第四步，Basic Theta*扩展到目标节点C1(父节点是B3)，中止搜索，通过从目标节点C1逐步检索父节点，提取出从开始节点到目标节点的最短路径是A4→B3→C1，如图6(d)所示。

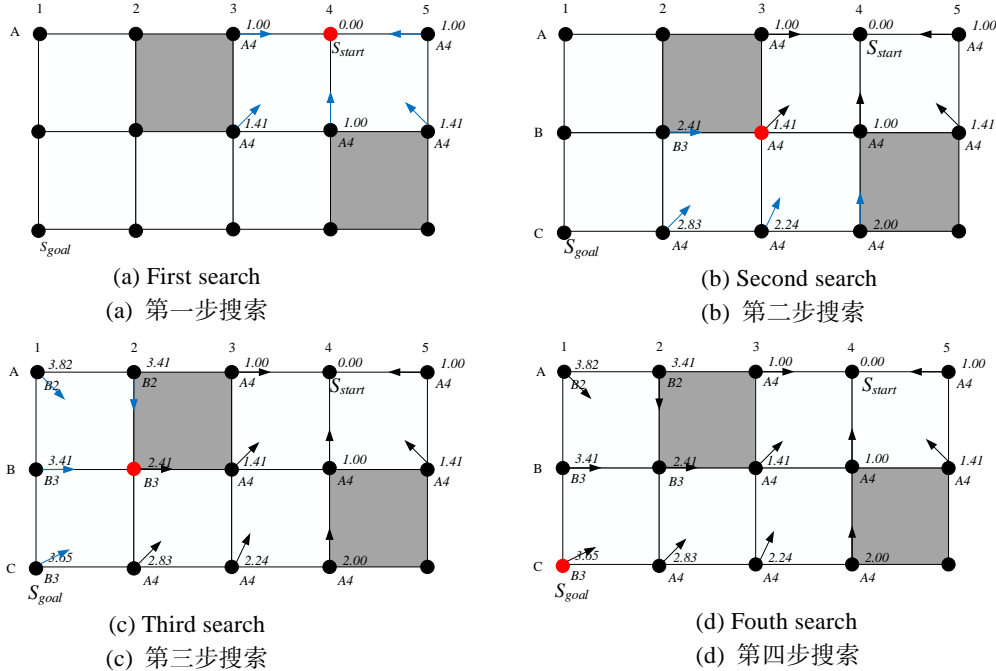


Fig.6 Example trace of Basic Theta*

图 6 Basic Theta*寻路实例

2.2 改进 LOS 算法

针对Basic Theta*算法使用的传统可视性检测算法存在检测误差的问题，通过改进传统视线算法中的判别指标，精确的确定栅格地图中两点间直线经过的所有栅格，之后对直线经过的栅格进行类型的判别，只有当直线经过的栅格都属于自由空间，则视线存在；反之，视线不存在。

同时为了克服基于网格顶点检测的传统视线

算法规划的路径存在不合理的部分，将视线检测算法的节点由网格顶点改为网格中心点。

2.2.1 搜索方式

首先规定搜索方向，假设我们需要从 S_{start} 这一点到 S_{goal} 画一条目标直线，两点之间的水平距离为 dx ，垂直距离为 dy 。从起点开始沿着某一坐标轴趋向终点的一侧进行搜索，该方向称为第一搜索方向。当 $dx > dy$ ，第一搜索方向为 x 坐标轴方向趋向

终点的一侧；当 $dx < dy$ ，第一搜索方向为y坐标轴方向趋向终点的一侧；当 $dx = dy$ ，第一搜索方向为 $y = x$ 直线趋向终点的一侧。与第一搜索方向垂直趋向终点的一侧称为第二搜索方向。

每执行一次搜索，按照第一搜索方向移动一个栅格位置，当到达满足第二搜索方向的位置，按照第二搜索方向移动一个栅格位置。对搜索方向的判断通过尾部距离来决定。

如图7所示，当前搜索节点为 (x_i, y_i) ，第一搜索方向为x轴，第二搜索方向为y轴。尾部距离表示当前搜索节点所代表栅格的边框中点在第二搜索方向上和目标直线的距离，图中表现为 d 。

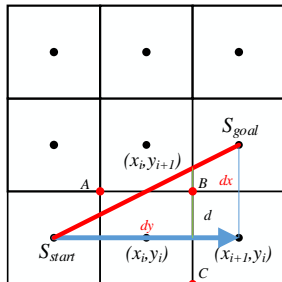


Fig.7 Tail distance

图7 尾部距离

在图7中，当前搜索节点为 (x_i, y_i) ，目标直线穿过该栅格时，当目标直线与栅格的交点正好落在B点时，此时尾部距离正好等于0.5，目标直线必定经过沿当前搜索栅格的 $y = x$ 方向上的下一个栅格；当目标直线与栅格的交点落在AB段时，此时尾部距离必定大于0.5，目标直线必定经过沿当前搜索栅格的y方向上的下一个栅格，即第二搜索方向上的下一个栅格；当目标直线与栅格的交点落在BC段时，此时尾部距离必定小于0.5，目标直线必定经过沿当前搜索栅格的x方向上的下一个栅格，即第一搜索方向上的下一个栅格；因此，根据尾部距离选择的搜索方向的判断阈值是0.5。

同时根据上述分析，该阈值是判别搜索方向是第一搜索方向还是第二搜索方向的临界阈值，因此选择其他阈值无法实现搜索方向的判别，达不到进行可视性检测的效果。

尾部距离判断原则：

(1)当 $d < 0.5$ ，说明直线下一步经过 (x_{i+1}, y_i) 点所在的栅格，不满足沿第二搜索方向的条件，按照规定的第一搜索方向即可到达。

(2)当 $d > 0.5$ ，说明直线下一步经过 (x_i, y_{i+1}) 点所在的栅格，满足沿第二搜索方向的条件，按照规定的第二搜索方向即可到达。

(3)当 $d = 0.5$ ，说明直线下一步经过 (x_{i+1}, y_{i+1}) 点所在的栅格，沿当前搜索栅格的 $y = x$ 方向上趋向终点的一侧即可到达。

在搜索过程中会维护一个列表Tile，该列表用于存放目标直线经过的栅格节点，通过尾部距离判断得到的搜索方向所到达的栅格，就是目标直线准确经过的栅格位置。

2.2.2 算法推导

算法的核心是对尾部距离的计算，通过判断原则，完成从起点到终点的搜索。如图8所示，假设 $dx > dy$ ，绿色方格表示当前搜索节点，x轴为第一搜索方向，y轴为第二搜索方向。

如图8(a)所示，当前搜索节点为起点D1，标记为绿色，将D1添加到列表Tile中。当前搜索节点的尾部距离为图中ab段，根据三角形相似原理可得， $d = 0.5 * (dy / dx)$ 。根据选择原则判断，下一步经过的栅格为D2，标记为绿色，将当前节点标记为蓝色，按照第一搜索方向即可到达。

如图8(b)所示，当前搜索节点为D2，将D2添加到列表Tile中。当前节点的尾部距离为图中ce段，根据图中几何关系可得，当前搜索节点的尾部距离等于在上一节点尾部距离的基础上增加了 dy / dx ， $d = 1.5 * (dy / dx)$ 。根据选择原则判断，下一步经过的栅格为C2，标记为绿色，将当前节点标记为蓝色，按照第二搜索方向即可到达。

如图8(c)所示，当前搜索节点为C2。将C2添加到列表Tile中。当前节点的尾部距离为图中cf段，C2节点是由D2节点沿第二搜索方向移动一个栅格单位所得，根据图中几何关系，该节点的尾部距离等于上一节点尾部距离减去一个栅格长度1，

$d = 1.5 * (dy / dx) - 1$ ，根据选择原则判断，下一步经过的栅格为C3，标记为绿色，将当前节点标记为蓝色，按照第一搜索方向即可到达。

如图8(d)所示，当前搜索节点为C3。将C3添加到列表Tile中。当前节点的尾部距离为图中 gh 段，根据图中几何关系可得，当前搜索节点的尾部距离等于在上一节点尾部距离的基础上增加了 dy / dx ， $d = (1.5 * (dy / dx) - 1) + (dy / dx)$ ，根据选择原则判断，下一步经过的栅格为B3，按照第二搜索方向即可到达。

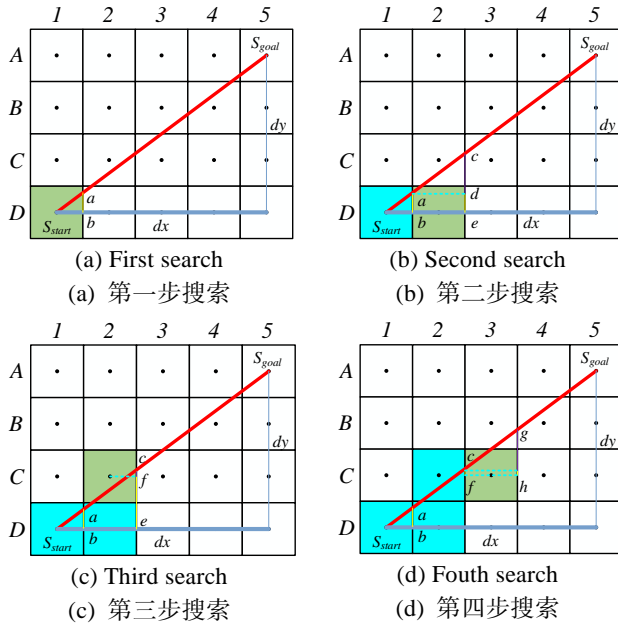


Fig.8 Derivation of improved line of sight algorithm

图 8 改进视线算法推导

按照这种搜索方式，最终到达目标节点，就可以找出目标直线经过的所有栅格。

从上述过程可以看出，尾部距离的计算可以一个偏差变量来代替，偏差初始值为 $-0.5 * (dy / dx)$ ，沿第一搜索方向执行一次，偏差增加 dy / dx ，沿第二搜索方向执行一次，偏差减少1，每一次的偏差值就等于当前搜索节点的尾部距离。最终通过对列表Tile中的栅格类型进行判断，只有当Tile列表中的栅格都为自由空间，才存在视线。当 $dx < dy$ 时，推导流程类似。

3 仿真结果及分析

3.1 Basic Theta* 算法仿真

为验证改进LOS算法确定栅格地图中两点间直线经过栅格的准确性，基于2.2.2节中对于算法原理的分析进行仿真实验，同时对比传统视线检测算法。改进LOS算法仿真结果如图9(a)所示，传统视线检测算法仿真结果如图9(b)所示。

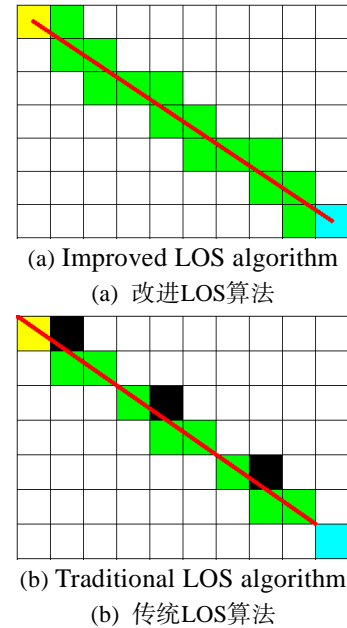


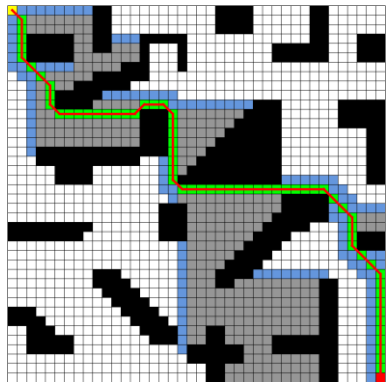
Fig.9 Line of sight algorithm simulation

图 9 视线算法仿真

图9中，黄色栅格表示起点，蓝色栅格表示终点，绿色栅格表示视线算法计算出来目标直线经过的栅格，黑色栅格表示传统视线算法漏检的栅格位置。传统视线算法是利用与栅格地图中两点间直线贴近的栅格位置，用离散的栅格序列来拟合直线，不包含目标直线经过的所有栅格位置，当障碍物的位置处在图9(b)黑色栅格位置处，因此传统视线算法所检测出的可视性结果是错误的。而改进LOS算法可以准确的计算出栅格地图中两点间直线经过的所有栅格，因此视线检测的结果是完全正确的。仿真结果表明，改进视线算法克服了传统视线检测算法存在误差的缺点，检测结果可以视线完全正确。

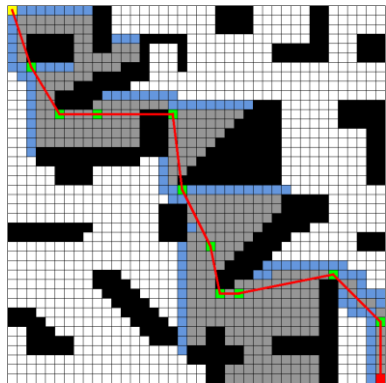
为了验证改进LOS算法对Basic Theta*算法的

效果,本文对传统的视线算法下Basic Theta*算法和改进的视线算法下的Basic Theta*算法进行仿真分析。选取了5个不同尺寸,障碍物密度相同的栅格地图,借助Pycharm环境,在CPU为i5-10400F,操作系统为Windows 10,主频2.9GHz,运行内存为16G的计算机上进行仿真。



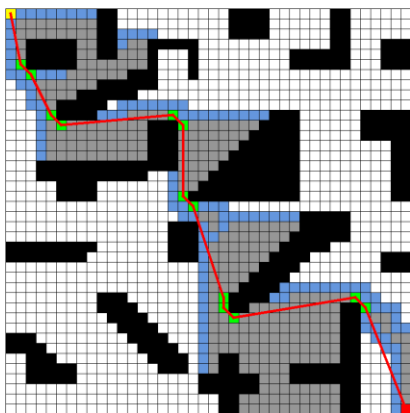
(a) Traditional A* algorithm

(a) 传统A*算法



(b) Basic Theta* algorithm

(b) Basic Theta*算法



(C) Basic Theta* algorithm based on improved LOS

(C) 改进LOS下的Basic Theta*算法

Fig.10 Simulation result of the A*algorithm and the improved

图 10 A*算法和改进 A*算法的仿真结果

如图10所示,两种算法在40*40栅格地图上的仿真实验,黄色栅格为起始节点,红色栅格为目标节点,黑色栅格为障碍物,蓝色节点是在Openlist中的节点,灰色节点为Closedlist中的节点,绿色节点在图10(a)中是路径经过的节点,在图10(b)和(c)是路径的上转折节点,转折节点越多表示路径的转折角越多,红色的轨迹是最终规划出的路径。图10(b)表示Basic Theta*算法使用基于栅格顶点的传统视线算法规划出来的不合理路径,图10(c)表示Basic Theta*算法使用基于栅格中心点的改进视线算法规划出来的合理路径。

在进行路径规划时,为了便于研究和开展实验,可以将机器人看做质点,并且假设转角处互为对角的栅格可以通过。图10(b)中,基于传统视线检测算法将当前栅格的左上顶点作为检测节点来替代机器人,图中黄色栅格的左上顶点和第一个经过的绿色栅格的左上顶点之间存在视线,从视线检测的角度来看是该路径是合理的;从机器人运动学的角度来看,考虑机器人的占地半径,可以将栅格视作实际机器人,从黄色栅格到第一个经过的绿色栅格的路线和障碍物区域存在交集,因此机器人无法顺利通过,该转折角无法实现,因此路径是不合理的。在图10(c)中,基于改进视线检测算法将当前栅格的中心点作为检测节点来替代机器人,从规划的路径来看,克服了传统视线算法规划的路径在考虑机器人占地半径时的不合理部分,直观上通过增加了转折格点的数量来实现了不合理路径的纠正,因此基于栅格中心点进行视线检测,可以提高路径在实际情况下的可实现性。仿真结果表明,基于网格中心点的检测方式可以提高规划路径的合理性,避免不合理路径的产生,符合实际的运行要求。

3.2 仿真结果定量分析

从图10(a)和10(b)中路径的对比可以直观上看出,传统A*算法规划出来的路径只能按照离散的45度方向来运动,轨迹被限制在网格方向上,转折角固定为135度,存在冗余的路径,单从算法的角度来看,路径的规划是符合要求的,但是从合理性考虑,路径过于机械死板,不具备灵活性;而通过Basic Theta*算法,在对A*算法的搜索过程中增加可视性检测算法之后,

规划出来的路径可以沿任意角度运动,不受网格的限制,是自然的最优路径。仿真结果表明,Basic Theta*算法能够视线任意角度的路径规划。

表1给出了传统A*算法和Basic Theta*算法2种算法在不同尺寸栅格地图中寻路的路径长度和搜索时间的对比,表中的数据均是10次试验的平均值。

Table1 Comparison between the A* algorithm and the Basic Theta* algorithm

表 1 A*算法与 Basic Theta*算法对比

地图规格	路径长度		搜索时间/ms		扩展节点数量/个	
	A*算法	Basic Theta*算法	A*算法	Basic Theta*算法	A*算法	Basic Theta*算法
10×10	16.4	14.90	0.204	0.204	27	27
20×20	30.5	27.75	0.206	0.206	43	44
40×40	71.9	65.30	1.317	1.330	364	341
80×80	120.8	109.80	6.614	6.723	835	840
100×100	161.5	147.10	15.659	16.376	1797	1839

图11(b)中的路径是Basic Theta*算法基于传统视线检测算法得出的结果,而图11(c)中的路径是Basic Theta*算法基于改进LOS算法得出的结果。可以看出,在图11(b)中,由部分路径和障碍物区域存在交集,这部分是属于不合理路径,也是基于传统视线检测算法的弊端;而在图11(c)中,基于改进视线算法规划的路径没有与障碍物交集的部分,符合机器人运动学规律,具备实际运行的有效性和合理性。

表2给出了Basic Theta*算法和改进Basic Theta*算法2种算法在不同尺寸栅格地图中寻路的路径转折格点数量和搜索时间的对比,表中的数据均是10次试验的平均值,可以清楚地看出,改进视线算

可以清楚地看出,Basic Theta*算法的路径长度在不同尺寸的地图中均比A*算法的路径长度明显减少,通过计算可得出,Basic Theta*算法比A*算法减少了约9%的路径长度。同时在搜索时间上,Basic Theta*算法相对A*算法时间的增量在 $10^{-3} \sim 10^{-1}ms$,对于实际应用的实时性影响不大。

法之后,Basic Theta*算法的路径长度在不同尺寸的地图中均比未改进视线算法之前的路径长度略微有所增加,通过计算可得出,改进视线算法之后的Basic Theta*算法比A*算法减少了约8%的路径长度;在搜索时间上,改进Basic Theta*算法相对Basic Theta*算法时间的略微有所减少,表明改进视线检测算法的运行效率略高。同时可以看出,改进视线算法之后,路径转折格点数量明显增加,这是因为需要通过增加路径的转折角数量来修正未改进之前存在的不合理的路径,会有一小部分路径成本的增加,不过总体上的路径缩减效果依旧很可观。

Table 2 Comparison between the Basic Theta** algorithm and the improved Basic Theta* algorithm

表 2 Basic Theta*算法与改进 Basic Theta*算法对比

地图规格	路径转折格点数量/个		搜索时间/ms		路径长度	
	Basic Theta*	改进Basic Theta*	Basic Theta*	改进Basic Theta*	Basic Theta*	改进Basic Theta*
10×10	7	7	0.204	0.204	14.90	15.088
20×20	7	10	0.206	0.205	27.75	28.029
40×40	12	15	1.330	1.320	65.30	66.248
80×80	25	39	6.723	6.665	109.80	111.136
100×100	40	61	16.376	16.154	147.10	148.58

综上,A*算法作为最短路径规划算法受到网格方向限制从而会产生部分冗余路径,达不到最优路径,而Basic Theta*算法能够有效克服这部分冗余路径的产生,多次实验平均削减率稳定在9%左右,算法稳定性优良。改进Basic Theta*算法不仅保证路径长度的平均削减率维持在8%,同时避免了传统Basic Theta*算法中不合理路径的产生,提高了算法的合理性。

4 结束语

本文针对传统A*算法在寻路过程中形成被限制在网格方向的路径,不具备路径的自然性,导致冗余路径产生等问题,提出一种任意角度规划的Basic Theta*算法,同时对传统Basic Theta*进行视线算法的改进。仿真结果表明,Basic Theta*算法不仅能实任意角度的路径规划,减少路径长度,而且改进Basic Theta*算法能够提高路径的合理性和可

实现性, 搜索的时间相较于传统Basic Theta*算法也有所降低。

本文也尚存在不足之处, 例如本文在对A*算法进行改进时, 通过在原先的搜索过程中增加视线检测过程, 势必会导致搜索时间的增加, 在表1和表2中可以看出相较于A*算法的时间增量维持在 $10^{-3} \sim 10^{-1} ms$, 在实验室和日常生活的运行过程中这个时间可以忽略。但是在特大地图中这个时间会随着地图的增加而有所增加, 因此在以后的研究工作中, 针对这个问题将每个进入到Openlist中的节点都要进行视线检测改为当节点被打开时才需要进行视线检测, 大大减少不必要节点的视线检测, 从而提高该算法的运行效率。

参考文献:

- [1] ZHAO X, WANG Z, HUANG C K, et al. Mobile Robot Path Planning Based on an Improved A* Algorithm [J]. ROBOT, 2018, 40(06):903-910.
赵晓, 王铮, 黄程侃, 等. 基于改进 A*算法的移动机器人路径规划[J]. 机器人, 2018, 40(06):903-910.
- [2] WANG H B, YI P H, ZHENG W, et al. Mobile Robot Path Planning Based on Improved A* Algorithm and Dynamic Window Method [J]. ROBOT, 2020, v.42(03): 92-99.
王洪斌, 尹鹏衡, 郑维, 等. 基于改进的 A*算法与动态窗口法的移动机器人路径规划[J]. 机器人, 2020, v.42(03): 92-99.
- [3] TAO Y, GAO H, REN F, et al. A Mobile Service Robot Global Path Planning Method Based on Ant Colony Optimization and Fuzzy Control[J]. Applied Sciences, 2021, 11(8):3605.
- [4] Zhao H R, Zhou H, Yang G Q. Research on Global Path Planning of Artificial Intelligence Robot Based on Improved Ant Colony Algorithm[J]. Journal of Physics: Conference Series, 2021, 1744(2): 022032-.
- [5] Guo X H, JI M J, ZHAO Z W, et al. Global path planning and multi-objective path control for unmanned surface vehicle based on modified particle swarm optimization (PSO) algorithm[J]. Ocean Engineering, 2020, 216: 107693-.
- [6] LONG Y, ZUO Z M, SU Y X, et al. An A*-based Bacterial Foraging Optimisation Algorithm for Global Path Planning of Unmanned Surface Vehicles[J]. Journal of Navigation, 2020, 73(6): 1247-1262.
- [7] CHENG Z, ZHANG C, AO L, et al. An Improved A* Algorithm Applying to Path Planning of Games[J]. Journal of physics. Conference series, 2020, 1631(1): 012068.
- [8] TIAN L. A Path Planning Method for Mobile Robot based on A* and Ant-Colony Algorithms[J]. Journal of Innovation and Social Science Research, 2020, 7(1).
- [9] JI X Y, FENG S, et al. Improvement and Fusion of A* Algorithm and Dynamic Window Approach Considering Complex Environmental Information[J]. Arabian Journal for Science and Engineering, 2021, : 1-15.
- [10] XIE L, XUE S F, ZHANG J F, et al. A path planning approach based on multi-direction A* algorithm for ships navigating within wind farm waters[J]. Ocean Engineering, 2019, 184(JUL.15):311-322.
- [11] DAI X L, LONG S, ZHANG Z W, et al. Mobile Robot Path Planning Based on Ant Colony Algorithm With A* Heuristic Method.[J]. Frontiers in neurorobotics, 2019, 13: 15.
- [12] FERGUSON D, STENTZ A. Using interpolation to improve path planning: The Field D* algorithm[J]. Journal of Field Robotics, 2006, 23(2):79-101.
- [13] DAVE, FERGUSON, ANTHONY, et al. Using interpolation to improve path planning: The Field D* algorithm[J]. Journal of Field Robotics, 2006.
- [14] PERKINS S, MARAIS P, GAIN J, et al. Field D* path-finding on weighted triangulated and tetrahedral meshes[J]. Autonomous Agents and Multi-Agent Systems, 2013, 26(3):354-388.
- [15] GAO B, XU D M, ZHANG F B. The Algorithm for the Dynamic Object based on Field D Algorithm and the Method of Path Extraction[J]. Fire Control & Command Control, 2010, 035(008):98-102.
- [16] WANG Q, DENG YUN Y U, JIA Y. An Improved Theta* Algorithm Based on Terrain Directional Traversability[J]. Chinese Journal of Space Science, 2016.
- [17] DANIEL K, NASH A, KOENIG S, et al. Theta*: Any-Angle Path Planning on Grids[J]. Journal of Artificial Intelligence Research, 2010, 39(1):533-579.
- [18] WANG Q, YU D Y, JIA Y. Risk Theta*:an Any Angle Path Planning Algorithm based on Terrain Risk[J].Journal of Deep Space Exploration,2014,1(04):269-274.
王琼, 于登云, 贾阳. Risk Theta*:一种基于地形危险度的任意航向路径规划算法[J].深空探测学报, 2014, 1(04): 269-274.
- [19] XIAO G B, YAN X H, A path planning algorithm based on improved Theta* for mobile robot[J].CAAI Transactions on Intelligent Systems, 2013(1):8.
肖国宝, 严宣辉. 一种基于改进 Theta*的机器人路径规划算法[J].智能系统学报, 2013(1):8.
- [20] MAO J, ZHANG H, LI H Y. Research of Anti-submarine Patrol Aircraft Route Planning Based on Lazy Theta* Alogrithm[J]. Ship Electronic Engineering, 2020, 40(12): 5.
毛杰, 张昊, 李海燕. 基于 Lazy Theta*算法的反潜巡逻飞机航路规划研究[J]. 舰船电子工程, 2020, 40(12): 5.
- [21] FARIA M, MARÍN R, POPOVIĆ M, et al. Efficient Lazy Theta* Path Planning over a Sparse Grid to Explore Large 3D Volumes with a Multirotor UAV[J]. Sensors, 2019, 19(1).
- [22] DERGACHEV S, YAKOVLEV K, PRAKAPOVICH R. A Combination of Theta*, ORCA and Push and Rotate for Multi-agent Navigation[C]//5th International Conference on Interactive Collaborative Robotics, St Petersburg, Russia, Oct 7-9, 2020.Springer, Cham, 2020: 55-66.



陈正泽 (1998-), 男, 江苏南京人, 江南大学硕士研究生, 主要研究方向为路径规划、轨迹跟踪。

CHEN Zhengze, born in 1998 year, M.S. candidate of Jiangnan University. He research interests include path planning and trace tracking.



彭力 (1967-), 男, 河北唐山人, 2002 年 6 月获得北京科技大学控制理论与控制工程专业博士学位, 现任江南大学物联网工程学院教授、博士生导师、中国人工智能学会会员和中国计算机学会会员, 主要研究领域为视觉物联网、行为识别、深度学习。CCF 普通会员。

PENG Li was born in 1967. He received Ph.D. in Control Theory and Control Engineering, University of Science and Technology Beijing, June 2002. Now he is a professor, PhD supervisor at School of Internet of Things Engineering, Jiangnan University, and the member of CAAI and CCF. His research interests include visual Internet of Things, action recognition and deep learning.



戴菲菲 (1988-), 女, 浙江临海人, 2014 年 6 月获得江南大学检测技术与自动化装置硕士学位, 目前于台州市产品质量安全检测研究院工程师, 主要研究领域为传感网络算法应用与研究, 智能检测方法研究与应用。

DAI Feifei was born in 1988. She received the M.S. degree in Detection technology and automation equipment from Jiangnan university in 2014 year. Her research interests include Application and re-search of sensor network algorithm, research and application of intelligent detection method.