



系统工程与电子技术
Systems Engineering and Electronics
ISSN 1001-506X, CN 11-2422/TN

《系统工程与电子技术》网络首发论文

题目：联合 A*与动态窗口法的路径规划算法
作者：汪流江，方德翔，李文刚，李玉玮，黄郡
收稿日期：2020-09-01
网络首发日期：2021-04-21
引用格式：汪流江，方德翔，李文刚，李玉玮，黄郡. 联合 A*与动态窗口法的路径规划算法. 系统工程与电子技术.
<https://kns.cnki.net/kcms/detail/11.2422.TN.20210420.1605.014.html>



网络首发：在编辑部工作流程中，稿件从录用到出版要经历录用定稿、排版定稿、整期汇编定稿等阶段。录用定稿指内容已经确定，且通过同行评议、主编终审同意刊用的稿件。排版定稿指录用定稿按照期刊特定版式（包括网络呈现版式）排版后的稿件，可暂不确定出版年、卷、期和页码。整期汇编定稿指出版年、卷、期、页码均已确定的印刷或数字出版的整期汇编稿件。录用定稿网络首发稿件内容必须符合《出版管理条例》和《期刊出版管理规定》的有关规定；学术研究成果具有创新性、科学性和先进性，符合编辑部对刊文的录用要求，不存在学术不端行为及其他侵权行为；稿件内容应基本符合国家有关书刊编辑、出版的技术标准，正确使用和统一规范语言文字、符号、数字、外文字母、法定计量单位及地图标注等。为确保录用定稿网络首发的严肃性，录用定稿一经发布，不得修改论文题目、作者、机构名称和学术内容，只可基于编辑规范进行少量文字的修改。

出版确认：纸质期刊编辑部通过与《中国学术期刊（光盘版）》电子杂志社有限公司签约，在《中国学术期刊（网络版）》出版传播平台上创办与纸质期刊内容一致的网络版，以单篇或整期出版形式，在印刷出版之前刊发论文的录用定稿、排版定稿、整期汇编定稿。因为《中国学术期刊（网络版）》是国家新闻出版广电总局批准的网络连续型出版物（ISSN 2096-4188，CN 11-6037/Z），所以签约期刊的网络版上网络首发论文视为正式出版。

联合 A*与动态窗口法的路径规划算法

汪流江¹, 方德翔¹, 李文刚^{1,*}, 李玉玮¹, 黄 郡²

(1. 西安电子科技大学通信工程学院, 陕西 西安 710071;

2. 国防科技大学电子对抗学院, 安徽 合肥 230037)

摘 要: 由于海洋环境的复杂性, 路径规划系统成为了无人艇智能系统中必不可少的一部分。为了解决 A*算法与动态窗口法(dynamic window algorithm, DWA)各自存在的缺陷, 本文提出了一种基于 A*与 DWA 的混合路径规划算法。首先使用 A*算法得到全局路径生成局部目标点, 再通过对局部目标点的不断更新使水面无人艇(unmanned surface vessel, USV)使用 DWA 航行到全局目标点完成路径搜索。仿真结果表明, 混合算法相较于 A*算法路径长度缩短了 10%, 且路径转折点减少了 67%。即本文所提出的混合路径规划算法的综合性能优于两种算法单独使用时的性能, 生成了一条从起始点到目标点的无差错且安全的平滑路径。

关键词: 水面无人艇; 路径规划; A*算法; 动态窗口法; 混合路径规划

中图分类号: TP 242 **文献标志码:** A

Path planning algorithm combining A* with DWA

WANG Liujiang¹, FANG Dexiang¹, LI Wengang^{1,*}, LI Yuwei¹, HUANG Jun²

(1. School of Communication Engineering, Xidian University, Xi'an 710071, China; 2. College of Electronic Engineering, National University of Defense Technology, Hefei 230037, China)

Abstract: Due to the complexity of the marine environment, the path planning system has become an indispensable part of the intelligent system of unmanned surface vessel (USV). In order to solve the defects of A* algorithm and dynamic window algorithm (DWA), a hybrid path planning algorithm based on A* and DWA is proposed. First, the A* algorithm is used to obtain the global path to generate a local target point. With the continuous updating of the local target point, the USV uses the DWA to sail to the global target point to complete the path search. Simulations show that, compared with the A* algorithm, the hybrid algorithm shortens the path length by 10% and reduces the turning point of the path by 67%. The performance of the proposed hybrid path planning algorithm is better than the performance of the two algorithms alone, and a correct and smooth secure path is generated from the starting point to the target point.

Keywords: unmanned surface vessel (USV); path planning; A* algorithm; dynamic window algorithm (DWA); hybrid path planning

0 引 言

近些年来, 无人系统受到了世界的广泛关注, 并且已经在海陆空 3 个方面获得了实质性的发展, 如无人车、无人机、无人航潜器及无人艇等^[1]。USV 是一种新型的智能化装备, 它是在无人操控的情况下可以自主航行的无人水面平台, 能够在海洋中完成低成本、大规模、

长时间的工程或科研任务, 因而在军事与民生方面都得到了宽泛的应用^[2]。

USV 航行环境十分复杂, 在受到岛屿、船舶等障碍物影响的同时还会受到风浪等因素的影响。因此 USV 对自主性和智能化的要求很高, 需要能够在航行途中完成与外部环境因素的交互, 其中最重要的部分就是路径规划^[3]。

路径规划是 USV 完成自动控制不可分割

的重要环节, USV 的路径规划系统, 详细可划分为全局路径规划与局部路径规划两部分。全局路径规划又称静态路径规划, 可以完成 USV 对静态障碍物的躲避。静态路径规划基于已知的海洋环境信息 (如电子海图等), 首先对全局环境信息进行抽象, 建立海洋环境信息的数字模型, 再通过特定的全局搜索算法进行路径搜索, 最终得到静态环境下的全局路径^[4]。局部路径规划又称动态路径规划, 可以实现 USV 对未知的动态障碍物的躲避。动态路径规划基于雷达及传感器探测得到的 USV 周围的环境信息, 首先对局部的环境信息进行抽象, 得到 USV 周围的最新障碍物分布情况, 并及时对海洋环境的数字模型进行更新, 再通过特定的局部算法对 USV 航行的方向与速度进行控制与调整, 最终可达到对动态障碍物的有效躲避。

在路径规划方面, 国内外许多学者已经进行了各种各样的研究, 并取得了优异的成果。巩敦卫^[5]等人为了了解决多地貌环境下的移动机器人路径规划问题, 提出了多目标微粒群优化算法, 但此算法不能完全适用于复杂的海洋环境。Kozynchenko^[6]等人针对动态障碍物状态的不确定性, 将人工智能算法应用于 USV 的路径规划问题。巩敦卫^[7]等人针对密集障碍物环境下存在控制偏差的机器人路径规划问题, 提出了一种基于凸包和微粒群路径规划算法, 但此算法不适用于存在动态障碍物的情形。张勇^[8]等人提出一种改进的多目标粒子群算法, 解决了具有不确定危险源的机器人路径规划问题, 但其对于动态环境下的问题没有进行相应的处理。于家斌^[9]等人为了了解决复杂环境下无人巡航船舶水质采样问题, 提出了一种基于 A* 算法与改进的 APF 方法的混合路径规划算法, 解决了传统的 APF 算法存在的目标不可到达和容易陷入局部极小点的问题。耿娜^[10]等人针对搜救领域的任务分配问题, 提出了一种基于粒子群算法的改进集中式算法, 该算法为路径规划算法提供了一个新的思路。王宁^[11]等人针对在复杂海洋环境下 (包括海岸线约束和水面约束) 航行的 USV, 提出了一种具有全局路径规划、避碰和常规校正功能的多层次路径规划器。

总体而言, 国外对于 USV 路径规划方面的探索展开的较早, 并且在传统的路径规划方

法的基础上进行了创新性研究, 如加入神经网络、强化学习等对算法进行改进^[12-16]。与此相比, 国内对于 USV 路径规划技术的研究与应用还相对较薄弱, 还有很多方面有待进行深入研究与探索。

1 环境模型

环境建模是路径规划中尤为重要的一步骤, 是后续环节进行下去的基础。首先获取环境信息, 其中包括由电子海图得到的海岸线、岛屿分布等静态信息, 以及由雷达、声呐等船载传感器获得的 USV 周围的移动障碍物分布等动态信息^[17]。接着对环境信息进行处理, 按照不同的策略选择合适的建模方法, 建立环境信息所对应的数学模型。该模型在路径规划中将完全代表真实的环境情况, 在此基础上会进一步地完成路径搜索、路径平滑等操作。因此环境建模的结果会影响到路径规划最终得到的航线质量以及航行过程中耗费的时间。本文综合考虑了多种环境建模的优缺点, 并与海洋环境进行匹配模拟, 最终选择使用栅格法进行海洋环境建模。并且就电子海图与栅格的对应、栅格粒度大小确定、障碍物轮廓处理等问题进行了进一步地说明。

1.1 栅格法建模

栅格法的原理是将整个空间按照一定的粒度拆分成一个个连续但不交叉的方格即栅格, 每一块栅格的状态代表着其所处位置的环境信息。通过拆分与环境对应将复杂且连续的问题简化为单一且离散的问题, USV 在整片海域的运动被拆分为相邻栅格之间的移动。详细步骤如下:

- (1) 根据标准粒度大小将整体空间拆分为相同的栅格;
- (2) 将各个栅格与实际环境空间中的位置区域相对应;
- (3) 对各个栅格的状态进行设置, 用于区分对应区域可否通过。

如图 1 所示, 确定栅格粒度大小后以此为依据对空间进行划分, 并与实际环境中的区域进行对应, 将障碍物标记为黑色部分表示无法通过, 白色区域即为可行区域。起点用 S 表示, 终点用 G 表示。其中对栅格位置进行标记时有两种方式。

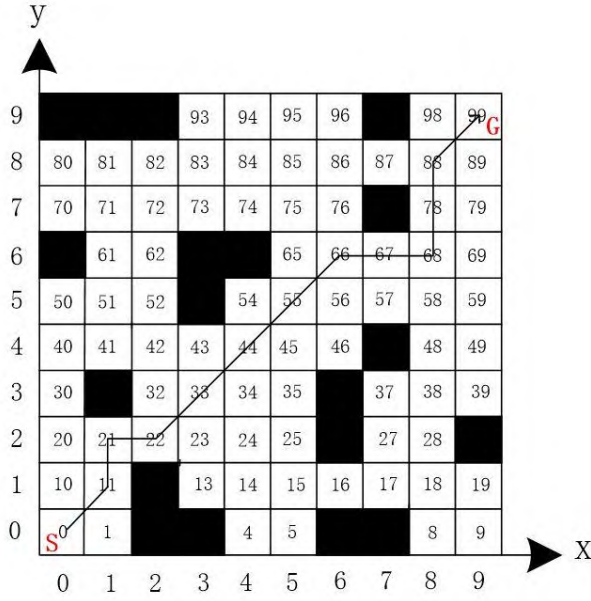


图 1 栅格法示意图
Fig. 1 Schematic diagram of grid method

(1) 坐标法: 以整体栅格图的两个边为基准建立横纵坐标轴, 将水平方向视为 X 轴, 竖直方向视为 Y 轴。则栅格的位置可以由坐标 (x, y) 来标记。

(2) 序列法: 按照一定的横纵规律以递增的方式对每一块栅格进行标记, 每块栅格都有自己所对应的编号。

上述两种标记方式可以互相转换, 其对应关系为

$$\begin{aligned} N &= x + n \times y \\ \begin{cases} x = \text{Mod}(N, n) \\ y = \text{Rem}(N, n) \end{cases} \end{aligned} \quad (1)$$

式中, x, y 为坐标法下的栅格横纵坐标, N 为序列法下的栅格编号, n 为栅格拆分粒度, Mod 为取余操作, Rem 为向下取整操作。

以图 1 为例, n 为 10, 最终得到的可行路线为图中实线所示, 其栅格序列按照坐标法和序列法分别可表示为:

$$\begin{aligned} M &= [(0,0), (1,1), (1,2), (2,2), (3,3), (4,4), \\ & (5,5), (6,6), (7,6), (8,6), (8,7), (8,8), (9,9)] \\ M &= [0, 11, 21, 22, 33, 44, 55, 66, 67, 68, 78, 88, 99] \end{aligned}$$

由上述两种不同的表示方法的表示形式, 本文在设计路径规划算法时, 为了表示方便, 因此采用坐标法来表示 USV 的运动环境。

栅格法的优点是大大减少了对复杂环境抽象处理的难度, 原理与实现都较简单。其缺点是划分栅格的标准难以把握, 若划分标准过

小会增加后续的搜索算法复杂度并且占用大量内存; 若划分标准过大会无法正确表现真实环境情况, 使得后续的搜索算法发生偏差并且无法得到实际的应用。因此在使用栅格法进行路径规划环境建模时, 需要权衡实际环境与搜索算法的要求等对划分标准进行选择。

1.2 栅格粒度确定

在栅格法建模过程中最为重要的一步就是栅格粒度的确定, 栅格的划分粒度在某种程度上决定了模型建立的优劣性。如若栅格划分粒度过小, 会使得后续的路径搜索过程难度加大, 搜索算法占用的空间和耗费的时间都会过大, 导致搜索算法无法达到优异的效果。如若栅格划分粒度过大, 会使得环境模型与真实环境空间差距过大, 路径搜索算法无法完成对真实海域中的障碍物的躲避。因此栅格粒度大小的确定对于环境模型的建立以及后续进行的路径搜索过程尤为重要。

栅格粒度大小的划分应该综合该区域的障碍物在整体环境中的占比, 如果占比过大应该将划分粒度适当减小, 以保证路径搜索算法的精度; 如果占比比较小可以将划分粒度适当增加, 以降低路径搜索算法的复杂度^[18]。本文最终选择了如图 2 所示的 60×60 的栅格地图, 此栅格粒度大小能在保证路径搜索算法的精度同时降低一定的计算代价。

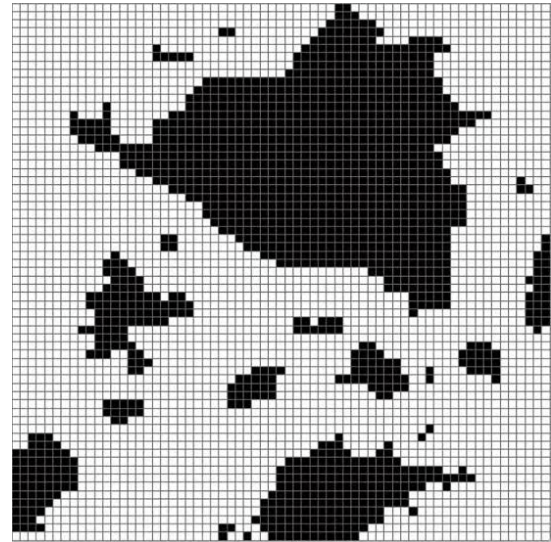


图 2 栅格粒度
Fig. 2 Grid granularity

2 A-STAR 算法的改进

2.1 A-STAR 算法概述

A-STAR 算法又称为 A* 算法, 是目前最为

普遍应用的一种启发式路径探索算法之一。其综合了经典的 Dijkstra 算法和 BFS 寻路算法的思想精髓。在解决了 Dijkstra 算法寻路速度较慢并且空间复杂度较高的问题的同时,也改进了纯粹的启发式寻路算法存在的无法保证路径最优化的缺点。A*算法提出了一个启发式函数即评价函数,评价函数包括起始节点到现时节点的实际消耗代价和现时节点到目标节点的预估消耗代价,将评价函数的计算值用于对当前节点一定邻域内的所有节点进行评估,作为下一步节点选择的评判标准。其评价函数可以表示为:

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

式中, $g(n)$ 表示从当前节点到目标节点 n 真实消耗的代价值,也被称为实际距离。 $h(n)$ 表示从当前节点到目标节点 n 预估即将消耗的代价值,也被称为估计距离。 $f(n)$ 即为综合考虑到已消耗的代价和可能消耗的代价的整体代价值。

评价函数中的 $g(n)$ 与 $h(n)$ 是一种相互制约关系。其中预估消耗代价 $h(n)$ 属于启发函数,是评价函数 $f(n)$ 中的重要部分,它的选择会直接影响到 A*算法的优劣性。在 A* 算法中,若预估消耗代价 $h(n)$ 过小而实际代价 $g(n)$ 过大,评价函数 $f(n)$ 中 $g(n)$ 的作用占主要部分,则算法会简化为 Dijkstra 算法,计算量会增加;如果实际代价 $g(n)$ 过小预估消耗代价 $h(n)$ 较大,则算法会简化为 BFS 算法,得到路径将无法保证最优性^[19]。

A*算法在实际路径搜索过程中的流程如图3所示。

2.2 堆结构优化 OPEN 列表

当在 USV 路径规划中应用 A*算法时,由于海洋环境的广阔,整体节点数会大大增加。若在 OPENLIST 中添加节点时,不考虑节点的 $f(n)$ 值大小,直接添加到数组的最后。那么每次选择扩展节点时,为了得到 $f(n)$ 值最小的节点,都需要对 OPENLIST 中的所有节点按照 $f(n)$ 的值进行大小比较,计算量会极大地增加。但如果添加节点时,就考虑节点的 $f(n)$ 值大小,将 OPENLIST 维持为一个有序的数组,那么添加节点与修改节点时计算量也会增加。为了平衡对 OPENLIST 列表的添加,修改,获取最小值等多种操作的效率,本文使用最小堆

结构来构建 OPENLIST 列表。

堆在数据结构中可以对应为完全二叉树,最小堆即在这颗完全二叉树中,任何一颗子树的最小值都是这颗子树的根节点。完全二叉树节点的存储可以由数组来完成,其中数组下标与完全二叉树的对应关系如图4所示。通过下标计算可以得到每个节点的子节点与父节点,设当前节点下标为 i , 则其父节点下标为 $(i-1)/2$, 左子节点下标为 $2i+1$, 右子节点下标为 $2i+2$ 。

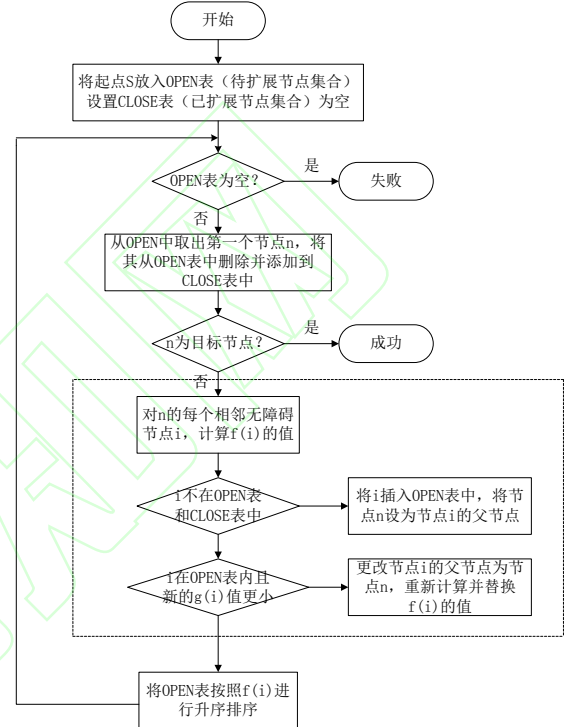


图3 A*算法流程图

Fig. 3 A* algorithm flow chart

20	30	40	33	36	50	42
0	1	2	3	4	5	6

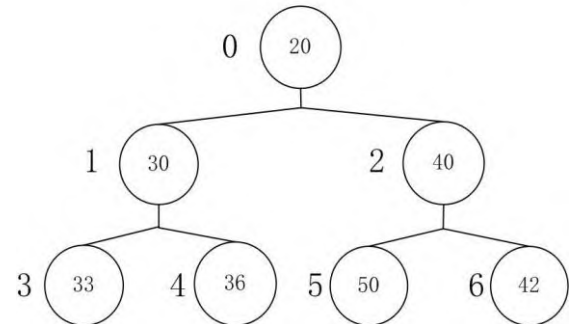


图4 堆结构的存储

Fig. 4 Storage of heap structures

为了方便操作我们将节点的 $f(n)$ 值作为节点的数值, 当需要将某节点添加到 OPENLIST 中时, 操作如下:

(1) 将新节点作为当前节点添加到 OPEN 列表的尾部;

(2) 通过完全二叉树与数组下标的对应关系, 计算出当前节点的父节点位置;

(3) 比较当前节点与其父节点的数值大小关系, 若当前节点值小于父节点其则交换二者位置;

(4) 向上逐层寻找父节点进行比较, 直到当前节点值大于等于其父节点值时停止。

当需要修改 OPENLIST 中某节点的数值, 最小堆结构会因为当前节点值的修改发生改变, 需要对修改节点的位置进行调整使来维持最小堆结构, 操作如下:

(1) 将修改节点作为当前节点, 找到当前节点两个子节点中数值小的记为节点 n ;

(2) 比较当前节点与节点 n 的数值大小关系, 若当前节点大于节点 n , 则将两者的位置交换;

(3) 向下逐层寻找子节点进行比较, 直到当前节点值大于其两个子节点值时停止。

当需要将 OPENLIST 中数值最小的节点删除时, 需要先将该节点从数组中剔除再重新调整最小堆的结构, 操作如下:

(1) 将堆顶节点与堆的最后一个节点交换位置;

(2) 将 OPENLIST 最后一个节点删除;

(3) 对堆顶节点的做位置调整以维持最小堆结构: 将堆顶节点作为当前节点, 找出当前节点的两个子节点中数值小的那个记为节点 n 。将节点 n 与当前节点的数值做大小比较, 若当前节点数值大, 则将两者位置交换。接着向下逐层寻找子节点进行比较, 直到当前节点值大于其两个子节点值时停止。

2.3 双向 A-STAR 算法

海洋的浩瀚与广阔使得路径搜索中初始点与目标点之间的距离很远, 在使用栅格法进行建模之后, 环境空间中的栅格数量往往是上千级别的。此时使用 A*算法从起始点开始向目标点进行单向扩展, 需要完成对途中大量节点的搜索, 在海洋障碍物环境较复杂时更会导致搜索算法占用空间与耗费的增加。为了提高 A*算法的搜索效率, 使其更适用于海洋环境, 本文对 A*算法进行改进, 使用双向 A*算法用于路径搜索过程。

双向 A*算法的详细步骤为:

(1) 建立两个 OPEN 列表和两个 CLOSE 列表, 其中 OPEN_正和 CLOSE_正用于存放正向搜索过程中的待检查节点和已经被扩展的节点, OPEN_反和 CLOSE_反用于存放反向搜索过程中的待检查节点和已经被扩展的节点;

(2) 将初始节点 S 添加到 OPEN_正中, 将目的节点 G 添加到 OPEN_反中。并将两个 CLOSE 列表都设为空;

(3) 正向搜索与反向搜索交替进行, 具体搜索过程与 A*算法步骤相同, 且每次搜索完成后确定的最优节点将作为下次搜索的目标节点;

(4) 当某方向搜索时确定的最优节点存在于另一方向的 CLOSE 列表中时, 说明正反两个方向的扩展重合, 搜索完成。

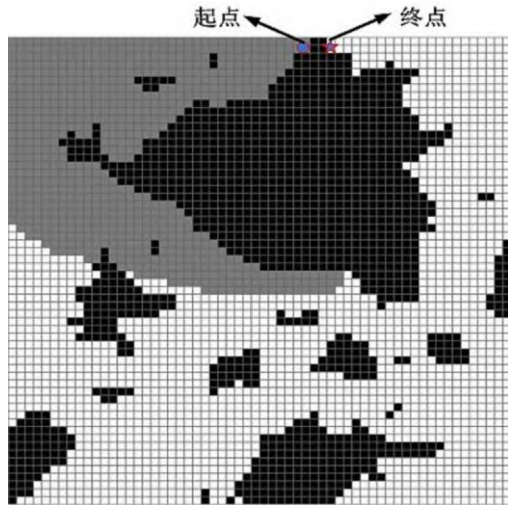
(5) 从重合节点开始, 向正反两个方法依次连接父节点, 最终得到的路线即为本次路径搜索的结果。

2.4 算法仿真

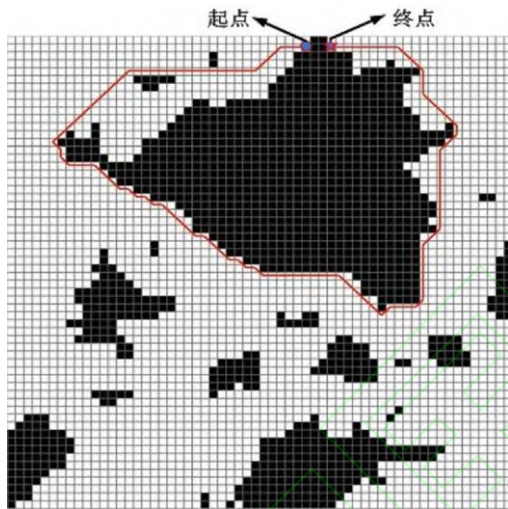
为了验证 A*算法与双向 A*算法在 USV 路径规划中的可行性, 并且比较两种算法的搜索效率, 在 MATLAB 2019a 中进行了相关仿真, 仿真场景为真实海洋环境经过建模之后得到的 60×60 的栅格地图。仿真设置了 3 组相同的初始点与目标点, 分别对 A*算法与双向 A*算法进行验证, 并对两种算法从路径长度、扫描节点、搜索时间等方面进行对比, 图片中仅展示一组的仿真结果, 表格对 3 组均进行了分析。

图 5 为起点为 (36,2) 终点为 (39,2) (图中箭头标记处) 的 A*算法仿真结果, 图(a)为双向 A*算法的搜索过程, 灰色部分为节点扩展过程, 图(b)为最终结果, 其中蓝色圆圈为设置的初始点, 蓝色五角星为设置的目标点, 红色路线即为搜索结果。由图中可以看出使用 A* 算法可以实现 USV 的全局路径规划, 完成对静态障碍物的躲避并且无差错的从起始点到达目标点。图 6 为起点为 (36,2) 终点为 (39,2)

(图中箭头标记处) 的双向 A*算法仿真结果, 图(a)为双向 A*算法的搜索过程, 灰色部分为节点扩展过程, 图(b)为最终结果, 其中蓝色圆圈为设置的初始点, 蓝色五角星为设置的目标点, 红色路线即为搜索结果。由图中可以看出使用双向 A*算法可以实现 USV 的全局路径规划, 完成对静态障碍物的躲避并且无差错的从起始点到达目标点, 并且能够在减少了检查节点的情况下得到与 A*算法相似的路径结果。



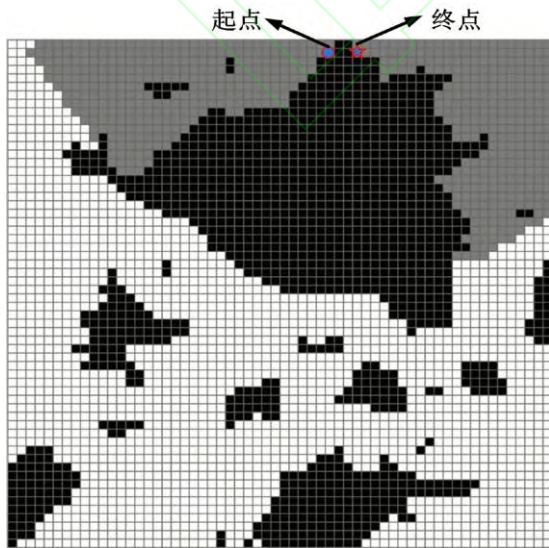
(a) 双向 A*算法的搜索过程
(a) Search process of bidirectional A* algorithm



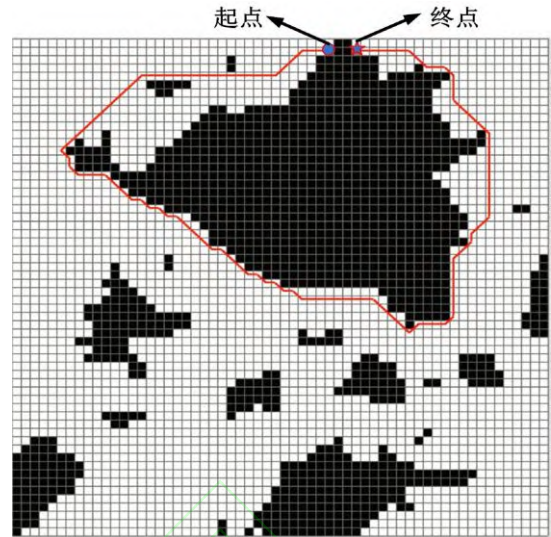
(b) 最终结果
(b) Final result

图 5 A*算法仿真图

Fig. 5 A* algorithm simulation diagram



(a) 双向 A*算法的搜索过程
(a) Search process of bidirectional A* algorithm



(b) 最终结果
(b) Final result

图 6 双向 A*算法仿真图

Fig. 6 Two-way A* algorithm simulation diagram

在起点为 (36,2) 终点为 (39,2) 的仿真设置下, 对 A*算法与双向 A*算法的仿真结果从路径长度、运行时间与占用内存等方面进行了比较, 结果如表 1 所示。

表 1 A*算法与双向 A*算法仿真结果对比

Table 1 Simulation results of A* algorithm and bidirectional A* algorithm are compared

算法	路径长度	运行时间(s)	OPEN 表搜索节点	CLOSE 表搜索节点
A*	134.04	10.95	132	1404
双向 A*	134.2	6.91	70	1143
A*	55.25	4.91	124	550
双向 A*	56.08	2.12	84	317
A*	150.34	14.25	160	2040
双向 A*	151.02	10.30	112	1623

由表 1 可以看出, 双向 A*算法相较于 A*算法在搜索时间上缩短了 60%, 节点列表空间减少了 25%。即双向 A*算法相较于 A*算法可以有效地缩短运行时间、降低扩展时的搜索节点数的, 同时在一定程度上能够保证所生成的路径的优劣性。在 USV 全局路径规划中使用双向 A*算法, 能够使其在耗费时间较短并且占用空间较少的情况下, 完成从初始点到目标点的无障碍安全路径获取。

3 动态窗口法

USV 的局部路径规划在其自主航行系统中是不可或缺一部分, 要实现 USV 在复杂海洋环境中的无障碍航行, 不仅需要完成对已知静态障碍物的躲避, 也需要完成对未知的动态

障碍物的躲避。USV 局部路径规划对于环境模型的精细度要求不高,主要通过雷达、声呐等船载传感器或高清摄像机等设备捕捉周围环境的障碍物信息(包括静态与动态障碍物),然后对 USV 当前航行的方向与速度进行调整,以实现周围障碍物的躲避,其保障了 USV 路径规划系统的实时性。本文使用 DWA 来完成 USV 的局部路径规划,并且在评价函数中加入了考虑海况级别的权重系数,当海况级别较大时降低速度的权重同时增加安全距离的权重,以此来保证 USV 实际航行的安全。

图 7 为 DWA 示意图,黑色船型表示 USV,灰色矩形代表障碍区域,各个曲线为符合 USV 运动约束的多组速度预估得到的多组路线轨迹。

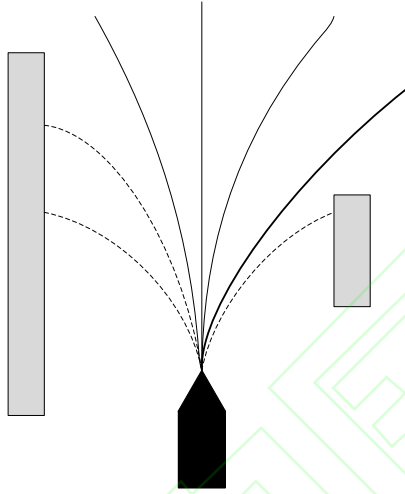


图 7 动态窗口法示意图

Fig. 7 Schematic diagram of dynamic window algorithm

如图中虚线所示,有 3 条轨迹在较近的距离内与障碍物发生了碰撞。因此选择最优轨迹时,首先舍弃掉这些不安全轨迹,再使用评价函数对剩余的轨迹进行评估,最终得到一个评价函数值最优的安全轨迹。

3.1 运动模型

DWA 的思想是对根据对轨迹的预测来选择下一时刻的速度,所以用于决定其行驶轨迹的运动模型尤为重要,本节在 USV 运动模型的基础上进行更详细的说明,以确定其位置与线速度、角速度、时间间隔等因子的关系。

假设 USV 的线速度 v_t 与角速度 ω_t 相互独立互不干扰,可通过改变 USV 的线速度与角速度来控制 USV 的运动,单位时间内一组速度 (v_t, ω_t) 对应着一段航行轨迹。为了方便轨迹

的计算与表示,我们假设 USV 在单位时间内作匀速直线运动,即单位时间内的加速度为零,但是在相邻的两个单位时间中存在加速度。可以看成在两个单位时间的间隔中,USV 的速度发生了改变更新了航行轨迹。这样建模易于对航行轨迹进行处理,USV 以固定的时间为单位对速度空间进行搜索并选择下一单位时间内的速度。

USV 在航行过程中可以进行前向与转向运动,设 USV 单位时间内的速度为 (v_t, ω_t) ,由于单位时间 Δt 的长度很短,在预估由当前单位时间到下一单位时间的轨迹时,可以将圆弧状的轨迹视为直线。若当前 USV 位置为 (x_0, y_0) ,方向角为 θ_t ,则单位时间后 USV 的位置与方向角可表示为

$$\begin{cases} x = x_0 + v_t \Delta t \cos(\theta_t) - v_y \Delta t \sin(\theta_t) \\ y = y_0 + v_t \Delta t \sin(\theta_t) + v_y \Delta t \cos(\theta_t) \\ \theta_{t+i} = \theta_t + \omega_t \Delta t \end{cases} \quad (3)$$

3.2 轨迹评价函数

传统的 DWA 中,评价函数考虑的参数为方位角、速度和距离。为了在外部干扰(例如风,浪,流)的作用下提高 USV 的航行性能,本文对 DWA 的评价函数进行了改进。在 DWA 的评价函数中加入了考虑海况级别的权重系数,以此在海况级别较大时降低速度的权重同时增加安全距离的权重。评价函数定义如下:

$$\begin{aligned} \max_{(u,r)} G(u,r) = \\ \alpha \cdot n_head(u,r) + k_1 \cdot \beta \cdot n_dist(u,r) \\ + k_2 \cdot \gamma \cdot n_vel(u,r) \end{aligned} \quad (4)$$

由式(4)可以看出最终评价值是综合 3 个评价函数得到的,其中包括方位角评价函数 $n_head(u,r)$ 、距离评价函数 $n_dist(u,r)$ 和速度评价函数 $n_vel(u,r)$ [20]。

式(4)中变量 α, β, γ 是 3 个评价函数的初始权重, k_1, k_2 为考虑海况系数的权重系数,定义如下:

$$\begin{aligned} k_1 &= 1 + \rho \cdot F \\ k_2 &= 1 - \eta \cdot F \end{aligned} \quad (5)$$

式中, F 是可以用风速计测量得到的海况级别, $(\rho, \eta) \in (0,1)$ 是可调的权重参数。由式(5)可以看出当海况较差时, k_1 会增加, k_2 会减少。因此评价函数中距离的权重将增加,速度的权重将减小,会优先选择安全距离大的且行进速度低的预测轨迹[20]。也就是说,随着海况级别的提高,USV 将以较慢的速度航行并具有更远

的安全距离。

为了保证航行轨迹的平滑性,需要对每个评价函数的值做归一化处理之后再进行加权运算:

$$\begin{cases} n_head(i) = \frac{head(i)}{\sum_{i=1}^n head(i)} \\ head(i) = 180^\circ - \theta \end{cases} \quad (6)$$

$$n_vel(i) = \frac{v(i)}{\sum_{i=1}^n v(i)} \quad (7)$$

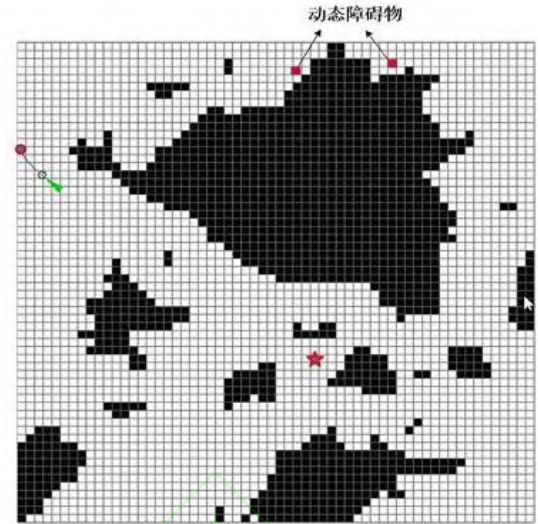
$$n_dist(i) = \frac{dist(i)}{\sum_{i=1}^n dist(i)} \quad (8)$$

式中, $head(i)$ 表示船头方向与目标点方向的夹角, $dist(i)$ 指的是预测轨迹的末端与障碍物的相距长度, $v(i)$ 指的是 USV 当前的航行速度。

3.3 DWA 算法仿真

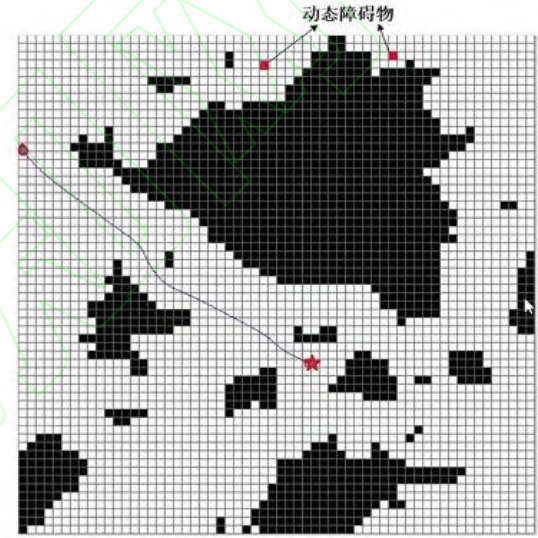
为了验证 DWA 在 USV 路径规划中动态避障的可行性,在 MATLAB 2019a 中进行了相关仿真,仿真场景为真实海洋环境经过建模之后得到的 60×60 的栅格地图。设置相同的起始点与目标点,分别在在路径搜索中受到动态障碍物影响和不受到动态障碍物影响的两种情况下进行仿真,并进行结果对比,验证了 DWA 算法在有动态障碍物环境中智能避障的优异表现。

图 8 所示为起点是 (1,14) 终点是 (35,40) 的 DWA 算法仿真图,图 8(a)中圆圈为设置的初始点,五角星为设置的目标点。图 8(a)中绿色范围即为搜索时的可行速度动态窗口,图 8(b)为最终结果。由图中可以看出动态窗口随着周围障碍物分布情况的改变在不断变化,蓝色路线为最终搜索结果。本次仿真所设置的动态障碍物活动区域为右上角区域,没有对此次路径达到干扰。由图中可以看出使用 DWA 算法实现了 USV 的路径规划,完成对静态障碍物的躲避并且无差错的从起始点到达目标点。



(a) 算法仿真图

(a) Algorithm simulation chart



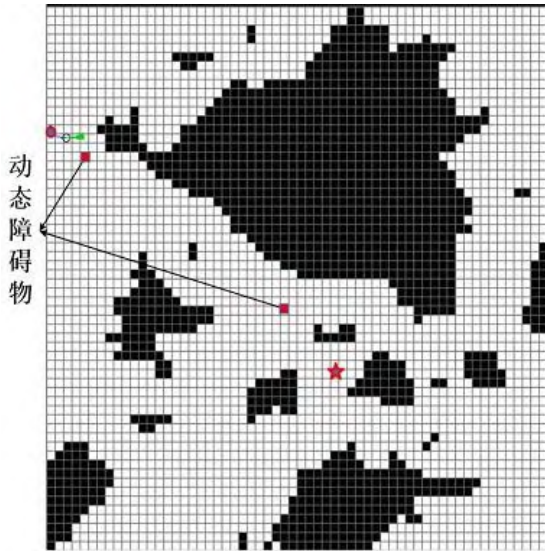
(b) 最终结果

(b) The last result

图 8 DWA 未受到动态障碍物影响仿真图

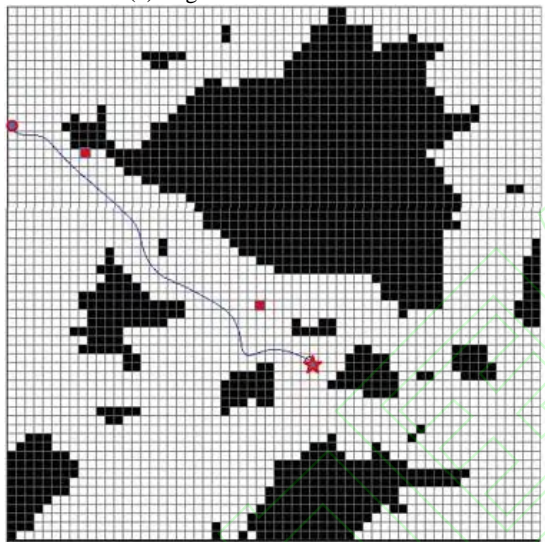
Fig. 8 DWA is not affected by the dynamic obstacle simulation diagram

图 9 的仿真场景与图 8 的仿真场景基本相同,不过本次仿真中,在 USV 会经过的区域内添加了两个分别沿着横纵方向移动的障碍物。将图 9(b)与图 8(b)进行对比可以发现由于动态障碍物的存在 USV 的路线发生了一定改变,说明了 DWA 算法在完成对静态障碍物的躲避的同时能够完成对动态障碍物的躲避,并且从起始点开始无差错地到达目标点。



(a) 算法仿真图

(a) Algorithm simulation chart



(b) 最终结果

(b) The last result

图 9 DWA 受到动态障碍物影响仿真图

Fig. 9 DWA is affected by the dynamic obstacle simulation diagram

虽然 DWA 算法可以完成对动态障碍物与静态障碍物的躲避,但是其作为局部路径规划算法存在容易陷局部最小值的问题,使 USV 无法到达最终目标,如图 10 所示,将起点设置为 (4,9) 终点设置为 (36,39) 时, DWA 算法陷入了局部最小点,路径规划失败。这是因为局部路径规划每次只处理周围的障碍物,没有全局的环境概念,因此容易前进到走不通的死角中。为了解决这个问题,本文提出了一种混合路径规划算法,将在下一节做详细说明。

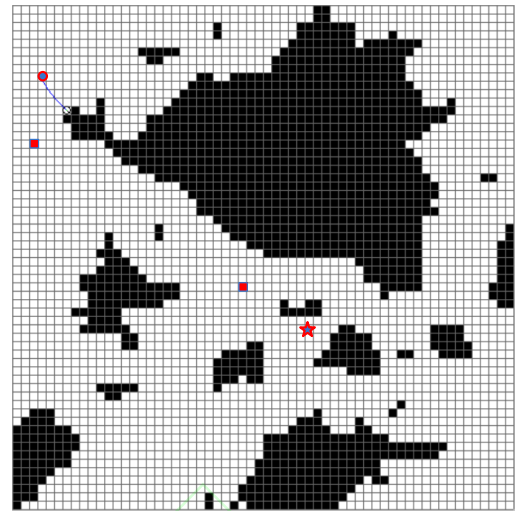


图 10 DWA 陷入局部最小点

Fig. 10 DWA falls into a local minimum

4 混合路径规划系统

4.1 混合路径规划系统建立

USV 在海面航行时的环境复杂,其中不仅包括岛屿、海岸线、暗礁等静态障碍物,还包括大型动物、其余船舶等动态障碍物,在进行 USV 路径规划研究时二者不可缺其一。基于 A*算法的全局路径规划只考虑了静态障碍物,没有考虑动态障碍物,导致 USV 可能与动态障碍物相撞。基于 DWA 算法的局部路径规划只考虑了当前位置周围的障碍物分布情况没有全局意识,导致 USV 的路径规划结果非最优且有陷入死角的危险。为了解决这些问题,本文提出了将上述两种算法相融合的混合路径规划算法。该混合算法不仅可以解决对静态、动态障碍物的躲避问题,还在能够保证路径最优的情况下提高了路径的平滑度,使其更适用于 USV 的实际航行情况。

图 11 所示为混合路径规划系统设计,该系统中主要包括全局路径规划器与局部路径规划器两个部分。首先将电子海图栅格化后得到全局地图,其中包括航行海域的静态障碍物分布情况,在全局路径规划器中使用双向 A*算法得到 USV 的全局路径;接着使用 USV 的感应器或雷达数据实时更新局部地图,并由全局路径与局部地图生成局部目标点,在局部路径规划器中使用 DWA 算法完成对局部海域内的障碍物躲避,通过局部目标点的不断更新使得 USV 最终到达全局目标点,并且规划出最优的轨迹。在 DWA 算法中优化评价函数,加

入与海况级别相关的系数。当海况较差时，距离的权重将增加，速度的权重将减小，USV 将以较慢的速度航行并具有较远的安全距离。

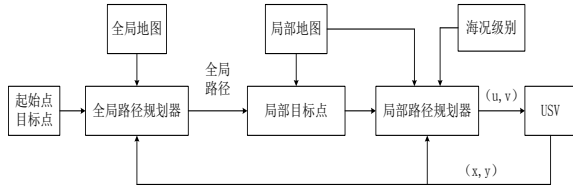


图 11 混合路径规划系统
Fig. 11 Hybrid path planning system

USV 混合路径规划系统的关键是全局路径规划与局部路径规划的融合，两者的融合关键部分在于局部目标点。如图 12 所示，首先由全局路径规划器在全局地图中生成 USV 的全局路径；其次将全局路径与局部地图范围的交点作为局部目标点；然后将局部目标点作为 USV 使用 DWA 算法进行路径规划的目标点；最终通过对局部目标点的不断更新使得 USV 达到全局目标点。

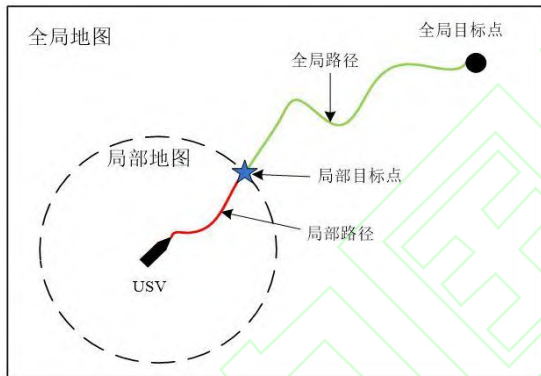


图 12 局部目标点示意图
Fig. 12 Local target point diagram

4.2 混合路径规划算法仿真

为了验证本文所提出的 USV 混合路径规划算法的可行性与优越性，在 MATLAB 2019a 中进行了相关仿真，仿真场景为真实海洋环境经过建模之后得到的 60×60 的栅格地图。在图 10 所示的 DWA 算法仿真结果中，由于缺少全局信息 USV 陷入了局部极值。为了验证混合路径规划算法对 DWA 的改进效果，本节的仿真条件与图 10 的仿真条件设置相同。

图 13 的仿真与图 10 的仿真中起点均设置为 (4,9) 终点均设置为 (36,39)，图中红色路线为改进 A*算法得到的全局路径，在此全局路径的引导下使用 DWA 算法进行路径搜索，搜索结果为蓝色路线。图 13 为混合算法的最终结果图，图中红色路线拐点较多且角度较

大，蓝色路线拐点较少且角度很小。



图 13 混合路径规划算法仿真
Fig. 13 Hybrid path planning algorithm simulation

表 2 所示为两种算法路径长度与拐点的对比与分析，混合算法相较于 A*算法路径长度减少了 10%，路径转折点减少了 67%。

表 2 混合算法与 A*算法仿真结果对比
Table 2 Comparison of simulation results between hybrid algorithm and A* algorithm

算法	路径长度	路径转折点
改进 A*算法	49	9
混合路径规划算法	44	3
改进 A*算法	25	7
混合路径规划算法	21	3

由图 13 和表 2 可以得出使用基于 A*算法与 DWA 算法的混合路径规划可以得到一条平滑且无冗余的无障碍安全路径。不仅解决了的 A*算法无法躲避动态障碍物的问题，同时改进了 DWA 算法容易陷入局部极值的缺陷，并且使用 DWA 算法对 A*算法的结果进行了平滑处理，使该算法更符合 USV 在海面实际航行时对转向角的限制。

5 算法时间复杂度分析

传统的 A*算法在最坏的情形下，即评价函数中实际代价函数的作用占主要部分，此时算法会简化为 Dijkstra 算法，计算量会增加，该算法在选择最短路径节点时需要访问所有的未标记节点，效率低下，整个算法的运行时间为 $O(n^2)$ ^[21]（其中 n 为节点的个数），而本文所提出的改进 A*算法的运行时间主要是由堆排序决定的，在排序过程中的时间复杂度不会超过满二叉树的高度 $\log(n)$ ，因此整个算

法时间复杂度为 $O(n \log(n))$ 。由于 DWA 的时间复杂度是线性的,因此本文提出的混合路径规划算法的时间复杂度取决于本文提出的改进的 A*算法。

6 结 论

USV 路径规划包括全局路径规划与局部路径规划,本文分别对其中的两个路径规划算法进行了改进,并且提出了一种混合路径规划算法。该算法使 USV 在能够完成对静态及动态障碍物的躲避,同时以平滑的路线完成航行,避免了 USV 由于转角过大而失去船体平衡。具体工作如下:

(1) 基于栅格法完成了对海洋环境的建模工作,对传统的 A*算法进行改进,包括使用堆结构优化 OPEN 列表以及使用双向搜索提高算法搜索效率,减少耗费的空间与时间。在同等条件下对改进的 A*算法与传统的 A*算法进行了仿真验证,结果表明改进 A*算法相较于传统 A*算法在搜索时间上缩短了 60%,节点列表空间减少了 25%。

(2) 基于 DWA 算法完成了 USV 的局部路径规划,实现了对局部环境中静态及动态障碍物的躲避。并且在其评价函数中加入了考虑海况级别的系数,使得海况级别增加时,评价函数中速度权重会降低而安全距离权重会提高,USV 会优先选择低速且安全距离远的轨迹完成航行。

(3) 将基于改进 A*算法的全局路径规划与基于 DWA 算法的局部路径规划相结合,提出了一种混合路径规划算法,首先使用 A*算法得到全局路径并生成局部目标点,再随着局部目标点的不断更新使 USV 使用 DWA 算法航行到全局目标点。通过仿真验证,混合算法相较于 A*算法路径长度缩短了 10%,且路径转折点减少了 67%。即该混合算法解决了 A*算法无法处理动态障碍物以及 DWA 算法容易陷入局部陷阱的缺陷,综合性能优于两种算法单独使用时的性能,最终生成了一条从起始点到目标点的无差错且安全的平滑路径。

参考文献

[1] 张伟,廖煜雷,姜峰,等.无人水面艇技术发展回顾与趋势分析[J].无人系统技术,2019,2(6):1-9.

ZHANG W, LIAO Y L, JIANG F, et al. Technology development review and trend analysis of unmanned surface craft[J]. Unmanned System Technology, 2019, 2(6): 1-9.

[2] LIU Z, ZHANG Y, YU X, et al. Unmanned surface vehicles: An overview of developments and challenges[J]. Annual Review in Control, 2016, 41: 71-93.

[3] 刘钰. 基于导航误差约束的水面无人艇路径规划方法研究[D]. 南京: 东南大学控制与科学工程, 2017.

LIU Y. Research on the path planning method of surface unmanned craft based on navigation error constraints[D]. Nanjing: Control and Science Engineering of Southeast University, 2017.

[4] WANG Y, LIANG X, LI B, et al. Research and implementation of global path planning for unmanned surface vehicle based on electronic chart[C]// Proc. of the International Conference on Mechatronics and Intelligent Robotics, 2018.

[5] 巩敦卫, 耿娜, 张勇. 多地貌环境下的移动机器人路径规划研究[J]. 控制与决策, 2012, 27(5): 708-712.

GONG D W, GENG N, ZHANG Y. Robot path planning in environment of many terrains[J]. Control and Decision, 2012, 27(5): 708-712.

[6] ALEXANDER I, KOZYNCHEV A. Applying the dynamic predictive guidance to ship collision avoidance: crossing case study simulation[J]. Ocean Engineering, 2018, 164: 640-649.

[7] 巩敦卫, 耿娜, 张勇. 密集障碍物环境下基于凸包和微粒群优化的机器人路径规划[J]. 控制理论与应用, 2012, 29(5): 609-616.

GONG D W, GENG N, ZHANG Y. Robot path planning in environment with dense obstacle based on convex hull and particle swarm optimization[J]. Control Theory and Application, 2012, 29(5): 609-616.

- [8] ZHANG Y, GONG D W, ZHANG J H. Robot path planning in uncertain environment using multi-objective particle swarm optimization[J]. *Neurocomputing*, 2013, 10: 172-185.
- [9] YU J B, DENG W, ZHAO Z Y, et al. A hybrid path planning method for an unmanned cruise ship in water quality sampling[J]. *IEEE Access*, 2019, 7: 87127-87140.
- [10] GENG N, MENG Q, GONG D, et al. How good are distributed allocation algorithms for solving urban search and rescue problems? a comparative study with centralized algorithms[J]. *IEEE Trans. on Automation Science and Engineering*, 2019, 16(1): 478-485.
- [11] WANG N, JIN X Z, JOOER M, et al. A multilayer path planner for a USV under complex marine environments[J]. *Ocean Engineering*, 2019, 184:1-10.
- [12] NAIR R S, SUPRIYA P. Robotic path planning using recurrent neural networks[C]// *Proc. of the 11th International Conference on Computing, Communication and Networking Technologies*, 2020.
- [13] RAMEZANLOU M T, AZIMIRAD V, ZAKERI M. Hybrid path planning of robots through optimal control and PSO algorithm[C]// *Proc. of the 7th International Conference on Robotics and Mechatronics*, 2019.
- [14] NOGUCHI Y, MAKI T. Path planning method based on artificial potential field and reinforcement learning for intervention AUVs[C]// *Proc. of the IEEE Underwater Technology (UT)*, 2019.
- [15] PANDA R K, CHOUDHURY B B. An effective path planning of mobile robot using genetic algorithm[C]// *Proc. of the IEEE International Conference on Computational Intelligence & Communication Technology*, 2015.
- [16] GOYAL J K, NAGLA K S. A new approach of path planning for mobile robots[C]// *Proc. of the International Conference on Advances in Computing, Communications and Informatics*, 2014.
- [17] 贾茂良. 基于电子海图的无人艇路径规划[D]. 天津: 天津理工大学, 2019.
- JIA M L. Unmanned boat path planning based on electronic chart[D]. Tianjin: Tianjin University of Technology, 2019.
- [18] SHI B H, SU Y X, ZHANG H J. Obstacles modeling method in cluttered environments using satellite images and its application to path planning for USV[J]. *International Journal of Naval Architecture and Ocean Engineering*, 2019, 11: 202-210.
- [19] 冯凯, 吉星, 杨昕. A*算法在自动驾驶车辆路径规划中的应用[J]. *汽车实用技术*, 2020, 45(22): 25-28.
- FENG K, JI X, YANG X. Application of A* algorithm in path planning of autonomous vehicles[J]. *Automobile Applied Technology*, 2020, 45(22): 25-28.
- [20] EIFFERT S, KONG H, PIRMARZDASHTI N. Path planning in dynamic environments using generative RNNs and Monte Carlo tree search[C]// *Proc. of the IEEE International Conference on Robotics and Automation (ICRA)*, 2020.
- [21] 梁波, 杨新民. 一种基于改进型 Dijkstra 算法的路线规划方法研究[J]. *信息化研究*, 2020, 46(2): 13-16.
- LIANG B, YANG X M. Study on a path planning method based on the improved Dijkstra algorithm[J]. *Informatization Research*, 2020, 46(2): 13-16.

作者简介

汪流江 (1997-), 男, 硕士研究生, 主要研究方向为高精度组合导航技术与传感器数据融合技术。

E-mail: wanglj@stu.xidian.edu.cn

方德翔 (1995-), 男, 硕士研究生, 主要研究方向为高精度组合导航技术。

E-mail: 498226792@qq.com

李文刚 (1980-), 男, 副教授, 博士, 主要研究方向为主要研究方向为宽带无线通信关键技术与卫星导航定位系统关键技术。

E-mail: wgli@xidian.edu.cn

李玉玮 (1995-), 女, 硕士, 主要研究方向为

智能路径规划技术。

E-mail: 465226738@qq.com

黄 郡 (1983-), 男, 副教授, 博士, 主要研究方向为无线网络信息安全技术。

E-mail: 151244199@qq.com

