

专业学位硕士学位论文

基于A*算法的无人船完全遍历路径规划研究 Research on complete traversal path planning of unmanned ship based on A*algorithm

作者姓名: 时天龙
学科专业: 交通运输
学 号: 1120211060
指导教师: 李广儒
学位类别: 交通运输硕士
培养单位: 航海学院
答辩时间: 2024 年 6 月 16 日

大连海事大学

Dalian Maritime University

摘 要

近年来, 由于海难事故频发和人们探索海洋世界的力度不断加大, 同时各国在海上军事方面的开展与研究也在不断加深。在这个背景下, 无人船技术的相关研究成为了当今海上智能化研究方向的热门, 受到了人们的高度关注和重视。而在无人船的相关技术中, 路径规划算法是无人船智能化的研究重点。它是无人船能否实现智能化、无人化的关键技术。完全遍历路径规划算法是路径规划算法的一种特殊形式, 在理论和实际应用上有重要意义。可以应用于水域边界巡逻、实行水上搜集任务等。

无人船在海上执行搜索任务时, 具有执行能力强, 救援速度快等优点, 当海上发生事故时, 如何在复杂的海洋环境中为无人船探寻一条安全、快速的可行路径以尽快实施救援成为了研究的热点。更好的路径规划算法可以使无人船用更合理、更高效的方式去探寻目标。

为了提高目前海上无人船的自主导航能力和机动性, 本文提出了一种在全局静态环境内基于 A*算法的无人船完全遍历路径规划算法。首先, 对传统 A*算法提出任意方向的 A*算法。并对传统 A*规划的路径存在与障碍物距离过近可能影响航行安全这一问题进行改进, 通过设置无人船行驶路径与障碍物之间的最小垂直距离大于等于一个无人船尺寸的距离来保障行驶的航行安全。当无人船遇到“死角”时, 利用水波纹算法, 从中心向四周搜索栅格地图中还未遍历的栅格作为终点, 到达此栅格后继续进行路径搜索。

对于局部路径规划算法, 本文采用 DWA 算法, 但是由于传统 DWA 存在一些问题。因此本文提出了一种改进 DWA 算法, 通过引入模糊控制方法, 使 DWA 算法与模糊控制相结合, 通过分析实时环境信息与无人船的运动状态来调整评价因子的权重, 提高了路径规划的性能和鲁棒性,

把改进后的 A*算法运用于双向 A*算法中, 当无人船在复杂的环境地图中不进行完全遍历时运用改进 A*算法进行路径规划。若进行完全遍历, 则采用内螺旋遍历算法对复杂环境地图进行完全遍历, 当无人船遍历陷入“死角”时, 运用改进双向 A*算法跳出死角, 并继续进行内螺旋完全遍历, 直至遍历完成。

最后, 通过一系列的仿真实验验证了所提出路径规划算法的有效性和可行性。实验结果表明, 无人船成功地避开了从起点到终点的障碍物。一方面, 缩短了无人船的路径距离和计算量, 根据无人船是否进行完全遍历路径规划, 采取不同的路径搜索算法, 不进行完全遍历时采取改进双向 A*算法进行路径搜索, 实验证明改进 A*算法在节点数目上比传统 A*算法更少, 规划路径长度更短。相比于传统 A*算法, 改进后的双向

A*算法路径更加平滑，转折点更少，当遇到“死角”时可以跳出“死角”，减少了重复遍历率，当进行完全遍历时，采用内螺旋完全遍历加改进双向 A*算法，无人船首先进行内螺旋完全遍历算法进行完全遍历，当无人船遍历陷入“死角”时，通过改进双向 A*算法跳出“死角”。相比于现有的完全遍历路径规划算法，当遍历路径长度相同的时候，该算法时间更短，算法性能得到改善。

关键词：无人船；路径规划；A*算法；环境建模；完全遍历算法

Research on complete traversal path planning of unmanned ship based on A* algorithm

Abstract

In recent years, due to the frequent occurrence of maritime accidents and the increasing efforts of people to explore the ocean world, countries have also deepened their research and development in maritime military. In this context, research on unmanned ship technology has become a hot topic in the field of intelligent research at sea, receiving high attention and importance from people. In the related technologies of unmanned ships, path planning algorithms are the focus of research on the intelligence of unmanned ships. It is the key technology for unmanned ships to achieve intelligence and unmanned operation. The fully traversal path planning algorithm is a special form of path planning algorithm, which has significant theoretical and practical applications. It can be applied to water boundary patrol, water collection tasks, etc.

Unmanned ships have the advantages of strong execution ability and fast rescue speed when performing search tasks at sea. When accidents occur at sea, how to explore a safe and fast feasible path for unmanned ships in complex marine environments to implement rescue as soon as possible has become a research hotspot. A better path planning algorithm can enable unmanned ships to explore targets in a more reasonable and efficient way.

In order to improve the autonomous navigation capability and maneuverability of unmanned ships at sea, this thesis proposes a complete traversal path planning algorithm for unmanned ships based on the A* algorithm in a global static environment. Firstly, propose an arbitrary direction A* algorithm for the traditional A* algorithm. And to improve the problem of traditional A* planning paths being too close to obstacles that may affect navigation safety, the minimum vertical distance between the unmanned ship's path and obstacles is set to be greater than or equal to one unmanned ship size distance to ensure navigation safety. When an unmanned ship encounters a "dead corner", the water ripple algorithm is used to search the grid map from the center to the surrounding areas that have not yet been traversed as the endpoint. After reaching this grid, a path search is carried out.

For the local path planning algorithm, this article adopts the DWA algorithm, but there are some problems with traditional DWA. Therefore, this article proposes an improved DWA algorithm, which combines the DWA algorithm with fuzzy control by introducing a fuzzy control method. By analyzing real-time environmental information and the motion status of

unmanned ships, the weight of evaluation factors is adjusted to improve the performance and robustness of path planning,

Apply the improved A* algorithm to the bidirectional A* algorithm, and use the improved A* algorithm for path planning when unmanned ships do not fully traverse complex environmental maps. If complete traversal is performed, the inner spiral traversal algorithm is used to fully traverse the complex environment map. When the unmanned ship traverses into a "dead corner", the improved bidirectional A* algorithm is used to jump out of the dead corner and continue the inner spiral complete traversal until the traversal is completed.

Finally, the effectiveness and feasibility of the proposed path planning algorithm were verified through a series of simulation experiments. The experimental results indicate that the unmanned ship successfully avoided obstacles from the starting point to the endpoint. On the one hand, it shortens the path distance and computational complexity of unmanned ships. Different path search algorithms are adopted based on whether the unmanned ship is fully traversed or not. When not fully traversed, an improved bidirectional A* algorithm is adopted for path search. Experimental results have shown that the improved A* algorithm has fewer nodes and shorter planned path lengths than traditional A* algorithms. Compared to the traditional A* algorithm, the improved bidirectional algorithm The A * algorithm has smoother paths and fewer turning points. When encountering "dead corners", it can jump out of them, reducing the repetition rate. When conducting complete traversal, the inner spiral complete traversal and improved bidirectional A* algorithm are used. Unmanned ships first perform the inner spiral complete traversal algorithm for complete traversal. When the unmanned ship encounters "dead corners" during traversal, the improved bidirectional A* algorithm is used to jump out of the "dead corners". Compared to existing fully traversal path planning algorithms, when the traversal path length is the same, this algorithm has a shorter time and improved algorithm performance.

Key Words: Unmanned ship; Path planning; A* algorithm; Environmental modeling; Complete traversal algorithm

目 录

1 绪论.....	1
1.1 研究背景及意义.....	1
1.2 国内外研究现状.....	2
1.2.1 完全遍历路径规划研究现状.....	2
1.2.2 无人船研究现状.....	2
1.2.3 路径规划算法研究现状.....	5
1.3 本文主要研究内容.....	6
1.4 本文框架.....	7
2 无人船运动模型与关键技术	9
2.1 无人船的数学模型.....	9
2.2 LOS 制导算法	10
2.3 基于栅格法的无人船环境建模	12
2.3.1 栅格法环境建模.....	12
2.4 海图栅格法环境建模.....	14
2.4.1 海图二值化处理.....	14
2.4.2 海图栅格化处理.....	15
2.5 本章小结.....	16
3 路径规划算法概述与改进	17
3.1 全局路径规划.....	17
3.1.1 A*算法	17
3.1.2 蚁群算法.....	20
3.1.3 路径搜索算法对比.....	22
3.2 局部路径规划.....	23
3.2.1 RRT 算法	23
3.2.2 DWA 算法.....	25
3.3 路径规划算法改进.....	27
3.3.1 改进 A*算法及算法实例.....	27
3.3.2 改进 DWA 算法.....	28
3.4 本章小结.....	30
4 无人船完全遍历路径规划	32
4.1 路径规划的核心问题.....	32
4.1.1 路径搜索问题概述.....	32
4.1.2 路径规划分类.....	32
4.2 完全遍历算法性能指标及“死角”情况.....	33

4.2.1 完全遍历算法的性能指标.....	33
4.2.2 “死角”问题	33
4.3 典型遍历算法.....	34
4.3.1 随机式遍历算法.....	34
4.3.2 螺旋式遍历算法.....	35
4.3.3 往返前进式遍历算法.....	35
4.4 基于改进 A* 的完全遍历路径规划.....	36
4.4.1 改进双向 A* 算法.....	36
4.4.2 改进双向 A* 算法的无人船完全遍历算法.....	38
4.5 本章小结.....	38
5 算法实验验证和结果分析	39
5.1 海图栅格建模及算法实验过程	39
5.2 算法对比实验.....	42
5.3 完全遍历路径规划实验.....	44
5.4 路径跟踪实验.....	44
5.5 本章小结.....	49
6 结论与展望.....	50
6.1 结论.....	50
6.2 展望.....	50
参考文献.....	52

图表目录

图目录

图 1.1 国外无人船型号	3
图 1.2 国内无人船型号	4
图 2.1 无人船运动模型	10
图 2.2 LOS 制导策略原理	11
图 2.3 栅格环境建模仿真图	13
图 2.4 海图二值化处理	14
图 2.5 海图栅格化处理	15
图 3.1 A*算法示意图	18
图 3.2 A*算法流程图	20
图 3.3 蚁群算法具体流程图	22
图 3.4 两种路径搜索算法示意图	22
图 3.5 RRT 算法搜索示意图	24
图 3.6 RRT 算法节点扩展方式	25
图 3.7 传统 DWA 算法搜索路径示意图	26
图 3.8 传统 DWA 算法搜索路径数据图	26
图 3.9 两种算法的路径示意图	28
图 3.10 改进 DWA 算法的路径示意图	30
图 3.11 权重因子自适应变化示意图	30
图 4.1 遍历陷入“死角”示意图	34
图 4.2 内螺旋式遍历算法示意图	35
图 4.3 改进双向 A*算法示意图	38
图 5.1 舟山群岛电子海图	39
图 5.2 舟山群岛栅格地图	40
图 5.3 实验过程示意图	41
图 5.4 传统 A*算法路径规划示意图	42
图 5.5 改进双向 A*算法路径规划示意图	43
图 5.6 内螺旋加改进双向 A*算法路径示意图	44
图 5.7 无人船行驶不同区域的航迹	45
图 5.8 A 区域规划航迹点航迹跟踪控制验证	47
图 5.9 B 区域规划航迹点航迹跟踪控制验证	48

表目录

表 3.1 两种路径搜索算法评价	23
表 3.2 RRT 算法伪代码	24
表 3.3 传统 A*算法与改进后的 A*算法数据分析	28
表 5.1 传统 A*算法与改进双向 A*算法数据分析	43
表 5.2 A 区域转向点及其坐标	45
表 5.3 B 区域转向点及其坐标	47

1 绪论

1.1 研究背景及意义

随着世界各国对海洋资源的不断探索和对相关领域科学技术的不断发展,无人移动平台在海陆空领域的相关应用也在快速增长^{[1][2][3]}。无人移动平台,即无人驾驶技术的应用,正在以前所未有的速度和广度渗透到我们的生活和工作中。从工业生产到军事战略,再到科学探索,无人移动平台正成为各个领域的重要组成部分。其中,无人机(Unmanned Aerial Vehicle, UAV)、无人车(Unmanned Ground Vehicle, UGV)在当今各国的军事和国防战略中大放异彩,因其具有众多功能而备受青睐。如监视、情报、搜索和救援、侦察等功能,使其在执行任务中发挥了重要的作用。当今无人移动平台的研究现状更是受到了各国的普遍关注。

近年来,随着国际社会对海洋事务的不断关注,无人海洋平台作为海洋领域的新兴技术,逐渐崭露头角。这一趋势与全球对海洋资源、环境和安全等议题的持续关注密不可分。无人海洋平台也逐渐成为各国研究的主要方向。例如,水面无人船(Unmanned Surface Vehicle, USV)在军事领域展现出显著潜力。其具备高度的隐蔽性和操作灵活性,使其成为海上侦察和监视的理想选择。此外,其小巧的体积、强大的负载能力、卓越的机动性以及相对较低的成本,使得水面无人船在各类任务中得以广泛应用。无人潜艇和无人水面舰艇的运用,可在不暴露人员位置的情况下执行海域监测任务,为军事行动提供关键情报和战术支援。在科学研究和资源勘探领域水面无人船也发挥着独特作用。面对海洋环境的复杂性和深度挑战,传统手段难以胜任。而配备各类传感器的无人海洋平台能够实现长时间、不间断的海洋调查和资源勘探,在广袤的海洋中为科学家提供宝贵的海洋数据和采集研究样本。

我们的国家是一个海洋大国,近年来,近海海洋污染事件和海上船只遇难事故频频发生,因此,需要对近海海域进行长期监测,以保护海上生命安全和维护海洋生态系统。同时,加强对关键海域的安全监控,包括打击非法采集渔业资源和应对未知船只的入侵。这些任务通常需要长时间、高强度的连续工作,采用传统的人力和物力会耗费大量资源。因此,迫切需要开发无人船替代人工操作,降低任务执行成本。我国近年来对无人海洋平台的研究已经取得了一定的成果,但是目前无人船大部分仍然属于半自主的无人海洋平台,或者只有在简单的海洋环境下才能实现基本的自主航行能力,例如在没有动态障碍物且水域环境已知的情况下,按照预先规划的路径去执行任务^{[2][3]}。由于海洋环境复杂且危险,水下环境无法准确预测,贸然使用无人船去执行科

考探索任务，很容易会与一些动态障碍物发生碰撞，使无人船的航行安全受到了极大的威胁。所以我们需要无人船具有鲁棒的导航和应对动态障碍物时候的控制策略。合理规划无人船的路线，以保证无人船可以安全在水域上自主航行显得至关重要。因此本文主要聚焦于 USV 在海面自主航行中的完全遍历路径规划技术的研究具有重大意义。

1.2 国内外研究现状

1.2.1 完全遍历路径规划研究现状

目前，各国对无人船完全遍历路径规划的研究通常可以分为四个步骤：首先，通过船上的环境感知装置将无人船所处的未知环境状况转为已知环境；第二步，通过对环境感知装置得到的已知环境进行建模，并生成适合无人船完全遍历路径规划算法运行的环境地图；第三步，在环境地图上运行无人船完全遍历路径规划算法；第四步，对无人船完全遍历路径规划得到的路径进行选择，选择最优路，并使无人船按照选择好的路径进行导航。在上述的步骤中，将未知环境转为已知环境是基础，步骤二和步骤三是整个算法运行的关键，同时，这也是目前各国对无人船完全遍历路径规划研究的重点，只有正确、合理的环境建模才能使最后规划的路径尽可能满足最优路径。

1.2.2 无人船研究现状

无人船的概念最早可以追溯到 20 世纪初，但其真正的起源要追溯到二战期间。在二战期间就已经出现无人船原型，在 20 世纪中期，正值二战结束，由于很多海湾的浅水区中存在未引爆水雷用人工排除危险性大且费时费力，各国纷纷开发和使用遥控船只并在其船头添加反水雷装置系统进行水雷排除等任务^[4]，这可以看作是无人船的雏形。然而，这些船只的控制相对简单，主要通过遥控操作。当时无人船作为武器应用在战场上，除了可以当作排除水雷的工具外，无人船当时还作为导弹或者军事武器在水面进行射击练习的靶艇^[5]。在 20 世纪 90 年代末期，由于美国海军范式转变的推动，极大的促进了无人船技术的进步和发展，为了更好的执行侦察和监视敌方的任务，美军研发了包括代号为“机器人-Roboski”和代号为“猫头鹰-Owl MK II”等在内的无人船，使无人船在当时的美国濒海战争和反恐任务中大放异彩^[6]。各国对无人船在战场上的众多优势得到认可，相继加大了对无人船的研究力度。

在进入 21 世纪后，随着卫星通信技术和海底通信技术、多传感器集成技术、自主导航技术、机器学习和深度学习等人工智能技术的发展，无人船的技术的限制也被相继攻克。无人船不再仅用于军事上，在民用无人船上的研发也变得炙手可热，各种军

用和民用产品络绎不绝。目前，世界上针对无人船技术研发的国家主要有：美国、俄罗斯、以色列、中国以及欧洲一些国家。国外无人船的研究和设计研发成果较为丰富，其中美国和以色列的成果最为凸出。

2002 年美国海下作战战略规划与发展研究室研发了一种即可远程遥控也可按照设定程序自动运行的高速无人船的核心系统——“卡车”，并把该核心系统应用于无人船上，该无人船被称之为——“斯巴克侦察兵”，该无人船可执行沿海地区反潜战、反水雷战、防御鱼雷、情报收集、监视海面信息和侦察等任务^[6]。2016 年美国军方研究出了专门为了反制潜水艇的无人艇——“海上猎人”^[7]，该无人艇的主要研发目的是解决美国海军近海反潜兵力的不足，具有自主航行、续航时间长、自动搜索跟踪、通信和网络功能等优点。2021 年基于“海上猎人”设计的无人艇“海鹰号”下海并投入使用，“海鹰号”具有更高的抗波稳定性和更强大的动力和载荷量，对抗恶劣海况环境优势更大。相比于其他无人艇，“海鹰号”还应用了人工智能技术，该技术可以在没有人工指导的情况下，使其进行巡逻，使用光学设备和雷达来避免撞到障碍物，如果发现目标，可以快速进行分类和识别。



(a) “海鹰”号



(b) “海上猎人”号

图 1.1 国外无人船型号

Fig 1.1 Foreign unmanned ship models

无人船军事化方面领先的国家还有以色列，以色列在无人船技术上的研究位于世界前列，仅次于美国。2003 年以色列国防部研发了最负盛名的“PROTECTOR”系列的 USV，该无人船具有良好的隐蔽性，能在不暴露自身身份的情况下执行一些关键任务，如侦察战场环境、战场保护、反恐、港口安全任务等，最大程度的降低了海军和船员的作战风险和提供了战场上环境信息^[8]。“PROTECTOR”采用远程遥控方式，最大航速能达到 40kn，该型号的无人船至今仍被菲律宾和美国军方采购并使用。2023 年以色列航空航天工业公司已研制出新型无人潜航器“蓝鲸”。该无人潜航器性能在世界处于领先水平，可以搭载最先进的声纳探测装备和加装执行各种任务所需的模块系统，两侧

设有侧舷声呐列阵和拖曳式声呐，可以搜集敌方潜艇的声纹信息和数据，并把相关坐标和信息发送给以色列海军的水面作战平台和潜艇。

其他国家在 USV 相关领域的研究也颇有建树，如英国为打击海盗，维护海上安全而设计的“Pacific 950”USV、法国为航道清扫、港口搜索而研发的“FDS-3”系列 USV、日本以巡逻警戒和海洋地理探测为目的而研发的小型高速系列 USV“Aquarius”。这些 USV 形态各异，适合执行的任务也不相同。随着人工智能技术、导航技术和相关技术的日益成熟，高度自主化和智能化是未来 USV 发展的主要方向。

我国在无人船这一方面相对于西方国家起步较晚，但是我国近年来也加大了对无人船的研究力度。随着国家对海洋科学研究和海洋经济的重视，无人船技术逐渐成为国内海洋科研和产业发展的重要方向。目前，我国对 USV 的研究主要集中在国内的一些顶尖高等学府、国家级科研机构院所以及相关企业。例如长期从事无人船及相关水下装备的研究工作的哈尔滨工程大学、在船舶工程及其自动化领域有着深入研究，包括无人船的设计和运营的大连海事大学、具有在海洋工程技术及无人船技术研究方面的强大实力的上海交通大学，还有武汉理工大学、天津大学、南京航空航天大学、中国船舶科学研究所、中国重工集团公司、海格通信等等^[9]。这些高等学府主要是侧重于基础理论研究和前沿技术探索。在 USV 技术领域研究通常包括技术集成、模拟仿真、路径规划、人工智能、自主控制等技术。而国家级科研机构院所通常承担着国家的重大科研项目，研究重点往往更加注重于应用导向和技术成果的转化，如针对特殊行业的需求，像海洋监测、水下勘探、军事应用、技术验证、标准制定等。而企业则是以市场需求为导向去研究 USV 相关技术，重点关注产品的研发、市场推广和商业化生产，企业的研究侧重点包括：产品开发、技术改进、商业应用模式探索，市场开拓等方面。



(a) “天行 1”号



(b) “瞭望者”号

图 1.2 国内无人船型号

Fig 1.2 Domestic unmanned ship models

1.2.3 路径规划算法研究现状

USV 在海上行驶的环境较为复杂, 易受障碍物的影响, 而路径规划技术是 USV 从起始点至终点寻求一条无碰撞安全路径的自主航行技术, 是无人船智能航行中不可或缺的一环^[10]。国内外学者对无人船路径规划系统已经研究出了很多相关算法, 并验证了其可行性。Seve 等人通过对传统 A* 算法进行优化, 将 A* 算法与博弈树方法相结合, 提出了一种混合路径规划算法^[11]。该算法在无人船进行路径规划时, 考虑了海浪对无人船航行时带来的偏振影响。通过基于最小极大值搜索算法来寻找无人船可能的偏航路径, 使规划出的路径更符合无人船的实际航行情况。但是当博弈树的数量很多的情况下会降低算法的路径搜索效率。巩敦卫等人为了解决移动机器人在多地貌环境下行驶的路径规划问题, 提出了多目标微粒子群优化算法^[12]。通过建立多目标优化数学模型, 并采用微粒群算法来进行模拟仿真, 验证了该算法的有效性, 但是该算法对无人船行驶在复杂水域内时并不能完全适用。传统 A* 算法的核心是算法中的启发函数, 通过设置不同的启发函数, 可以对算法进行改进, 以满足不同环境下的路径规划需求。而为了解决 A* 算法过度依赖启发函数的问题, 陈超等人基于可视图算法对 A* 算法进行改进^[13], 通过分析无人船行驶空间点与环境地图中的障碍物的位置关系, 使用启发式搜索方式克服了传统可视图法灵活性差的问题, 减少了算法规划路径时间, 提高了搜索效率和无人船对复杂海洋环境中的适应能力。郭峰针对在动态环境地图中运动的无人船提出了反向偏心与安全性圆形的两种动态碰撞膨化方式, 并设计了基于运动速度模型和改进蚁群优化算法的 USV 动态避碰方法, 但是没有考虑到当无人船行驶速度较快时风浪流对无人船行驶时的影响^[14]。Wang.N 等人提出了一种基于 A* 算法与动态窗口法相结合的融合算法, 该算法通过考虑无人船在不同情况下的行驶状态和操纵需求, 通过对无人船行驶路径进行平滑处理, 并删除了无人船行驶过程中的重复路节点, 减少了算法的储存空间。同时, 此混合算法不仅解决了无人船面对动态障碍物与静态障碍物时的避碰问题, 在保证最优路径的情况下使算法规划的航行路径更贴合实际航行情况, 具有很强的实用性^[15]。Singh 等人利用 Dijkstra 算法在模拟海上环境地图中对无人船进行路径规划, 考虑了洋流对无人船路径规划的影响, 将无人船的周围不规则障碍物包围起来定义为圆形, 启发式函数中考虑了最优路径曲率限制, 为无人船在环境地图中规划了一条安全可行最佳路径^[16]。王文明针对无人船自主靠泊过程中的全局路径规划、局部路径规划避障等方面进行研究, 通过在靠泊点和泊位点之间添加一个过渡点将其分为远端和近端路径规划问题。通过对电子海图进行构建正六边形的栅格地图, 修改传统跳点搜索算法的剪枝规则和跳点搜索策略, 提出了新的正六边形栅格跳点搜索算法, 解决了无人船远端路径规划问题^[17]。Kumari.M 等人将机器人运动历史数

据进行处理, 将其作为一个概率倍增系数。将蚁群算法中的信息素更新方式进行改进, 确保算法能够跳出局部最优解^[18]。曾凡明等人对于蚁群算法存在不能利用系统反馈信号, 迭代效率低下, 规划出的路径有时不是最优路径等问题进行研究, 通过使用遗传算法来对蚁群算法中的参数进行赋值, 加快蚁群算法的运行效率, 最后通过蚁群算法进行求解, 确保了路径解的正确性^[19]。沈童针对粒子群算法容易陷入局部极值且某些情况下优化速度过慢的问题, 将模拟退火算法与粒子群算法相结合并对其做出改进, 采用 Hammersley 点集代替传统粒子群算法随机初始化, 使粒子分布更加均匀, 算法性能更加稳定。同时采用基于种群分布密度的自适应惯性权重取代传统的线性惯性权重, 优化算法搜索精度与速度。设计出了一种将速度障碍法与基于模拟退火的自适应粒子群算法进行结合的无人船自主避障策略, 并进行仿真实验验证了算法的可行性与鲁棒性^[20]。庄佳园等人对传统 RRT 算法存在搜索速度慢, 无法实时进行路径规划等缺点进行研究, 提出了一种基于 RRT 的全局路径规划算法。通过对雷达显示的环境进行建模, 在已有的随机树扩展中引入抑制因子, 限定转角和距离启发信息, 改进生长点和探索点的选择, 提高了算法速度。删除了路径上的多余航行节点, 对规划的路径进行了平滑处理, 提高了算法的求解速度。但是在航行环境较为复杂、障碍物较多的情况下算法仍有可能陷入局部最优^[21]。

1.3 本文主要研究内容

本文对无人船路径规划的研究主要有以下两个方面: 第一, 由于环境的复杂性所以对航行环境的建模无法形成统一标准且建模时间过长。第二, 目前完全遍历算法存在的完全覆盖率低, 重复遍历率高, 复杂环境时容易陷入“死角”等。对于这两个问题, 本文希望通过改进后的完全遍历路径规划算法可以使无人船在栅格化后的电子海图内进行完全遍历。因为当前的很多无人船路径规划算法都是在设置好的环境地图内寻求一条无碰撞的安全路径, 但是在很多场景下, 需要对地图进行完全遍历路径规划。完全遍历路径规划算法的最早提出是为了基于为了完全解决区域内的森林着火这一目的需求, 通过对区域内的地图进行完全遍历, 搜寻所有的可能着火点并进行扑灭。基于这一目的, 类比海上救援, 本文提出基于 A*算法的完全遍历路径规划算法, 以便更好的解决上述的两个问题。本文的主要研究工作和创新点说明如下:

1.对无人船环境建模方法的研究。由于栅格法环境建模相比于无人船通过传感器自主确定环境边界建立环境模型的方法省时、省力、过程相对简单。所以本文通过栅格法进行环境建模, 首先对无人船航行区域的电子海图进行栅格化处理, 确定好比例尺进行二值化, 并将二值化后的海图与真实电子海图进行对比, 使其更加符合真实航行

环境地图。将处理后的二值化海图进行栅格化处理，通过选取合适的分辨率和确定地图的边缘坐标信息，以及每个栅格所代表的大小范围，最后将无人船放置在栅格化后的电子海图中进行路径规划。

2.对无人船全局路径规划算法和局部路径规划算法存在的一些问题进行研究。针对全局路径规划中的传统 A*算法存在的一些缺点：搜索路径经常出现转折点较多，且不符合无人船的真实运动情况，搜索路径不够平滑，规划路径与障碍物距离过近存在危险等情况，对传统 A*算法进行改进，提出了任意方向的转向改进和其他一些优化处理。对于局部路径规划算法，对于局部路径规划算法中的传统 DWA 存在需要人为手动操作设置评价函数权重等参数，导致算法可能在某些地图中的路径规划结果并不是很理想等问题，提出了一种改进 DWA 算法，通过引入模糊控制方法，使 DWA 算法与模糊控制相结合，通过分析实时环境信息与无人船的运动状态来调整评价因子的权重，可以根据实际的情况进行动态调整，以适应不同的场景和任务需求。

3.对无人船完全遍历路径规划算法的研究，针对完全遍历算法存在的搜索效率低，普适性差的问题，本文提出了一种基于改进 A*算法的双向 A*算法与内螺旋完全遍历算法结合的算法。该算法通过分析是否进行完全遍历对栅格地图进行路径搜索和完全遍历两种情况，当不进行完全遍历时，通过改进双向 A*算法可以快速搜索出路径，当无人船遇到“死角”时，可以通过搜索最近未遍历栅格进行跳出。若进行完全遍历，则通过内螺旋完全遍历算法对栅格地图进行完全遍历，如果遍历过程中遇到“死角”，则采用改进双向 A*算法跳出“死角”，继续遍历。该算法通过考虑路径搜索中与障碍物的安全距离、无人船航行时的路径节点数目、转弯次数、算法搜索时间使得算法规划出的路径更适合。

1.4 本文框架

本文共六章，具体如下：

第1章：阐述本课题的研究背景及研究无人船路径规划技术的意义，对国内外和无人船路径规划相关的重要技术和各国研究的无人船代表类型产品进行了介绍，重点分析了国内外研究人员对无人船路径规划算法的研究。最后简单阐述了论文的研究内容和主要创新点，并给出论文的整体结构框架安排等内容。

第2章：分析了无人船在航行环境内的运动，把无人船在海面上的运动看作二维平面内的运动，采取船舶三自由度运动学模型作为无人船的运动模型，介绍了无人船进行跟踪控制的常用算法 LOS 制导算法基本原理。同时对无人船栅格环境建模的基本流程和二值化、栅格化的过程进行了叙述。

第3章：对无人船的路径规划算法进行了概述，分别叙述了无人船全局路径规划算法和局部路径规划算法各自适应的场景和各自的优缺点。全局路径规划采用 A*算法。同时，对传统 A*算法进行详细阐述，介绍了算法的基本原理和概念，并对传统 A*算法存在的不足之处进行改进，提出了任意方向的转向改进和其他一些优化处理。对于局部路径规划算法，采用 DWA 算法，但是由于传统 DWA 算法存在一些不足，提出了一种改进 DWA 算法。通过引入模糊控制方法，使 DWA 算法与模糊控制相结合，通过分析实时环境信息与无人船的运动状态来调整评价因子的权重，提高了路径规划的性能和鲁棒性，可以适应不同的场景和任务需求。

第4章：首先探讨了路径规划的核心问题和“死角”这一特殊情况，分析对比了几种遍历算法，选择螺旋式遍历算法作为本文的遍历算法。在第三章改进 A*算法的基础上对双向 A*算法进行改进，最后将改进双向 A*算法与内螺旋完全遍历算法相融合，提出了一种改进双向 A*算法的无人船完全遍历路径规划算法。

第5章：本章通过对无人船行驶过的路径进行路径跟踪实验验证了第3章和第4章所提出的基于 A*算法的改进双向 A*算法的完全遍历路径规划算法的有效性。首先提出了实验流程方案，介绍了所使用的电子海图，并通过第2章方法处理成栅格地图，确定好边缘区域坐标和无人船的起始位置坐标。通过在栅格地图中进行实验，验证了改进双向 A*算法的优点和所提出完全遍历路径规划算法的可行性和有效性。

第6章：结论与展望。

2 无人船运动模型与关键技术

无人船在海面时的航行位置、航行状态、船艏向，航向等航行信息均由自身的传感器提供，无人船自身的传感器对无人船的航行环境进行实时探索监测并将其反馈给信息处理系统。无人船数学模型是准确表达无人船运动特征，并用于无人船进行路径跟踪^[22]研究的基础。LOS (Line Of Sight) 制导算法是无人船进行路径跟踪的常用算法之一^[23]，在无人船跟踪中起着关键作用。而栅格环境建模是无人船路径规划算法规划路径的前提，只有合理的对无人船航行环境进行建模，才能保证规划的路径最大程度上接近真实航行路径。

因此，本章将从这几个部分对无人船进行分析研究。

2.1 无人船的数学模型

无人船运动模型的建立可以用来对无人船的运动规划和所处位置进行预测，无人船在海面上的运动可以视为在二维平面内的运动，即不考虑无人船水下部分的运动状况。大多数船舶在海上运动时，当船舶发生垂荡、纵摇和横摇运动对船舶在水平面内的运动影响不大，故本文不考虑船舶在其自由度方向上的振荡运动。在船舶运动和控制领域，人们首先关心的是在二维平面内无人船的航向角 ψ ，以及在X、Y轴上的位移量 X_0 、 Y_0 。所以 ψ 、 X_0 、 Y_0 只取决于U(前进速度)、V(横移速度)、r(艏摇角速度)。即在本文中把无人船的运动问题视为三自由度模型。

无人船在平面坐标系X、Y轴上的位移增量可以表示为：

$$\begin{cases} X_0 = \Delta S \times \cos \Delta \theta \\ Y_0 = \Delta S \times \sin \Delta \theta \end{cases} \quad (2.1)$$

在上式中， X_0 为无人船在X轴上的位移增量， Y_0 为无人船在Y轴上的位移增量， ΔS 为无人船的位移增量， $\Delta \theta$ 为无人船航向与X轴的夹角。

无人船的航向角的范围为 $[-180^\circ, 180^\circ]$ ，正北方向为 0° 。无人船的横切宽度设为 W_s ，无人船与岸边的距离为Length米。无人船的运动模型如图2.1示。图中蓝色块为无人船，无人船此时的航向角为 ψ ，经纬度坐标 $O(latX_0, lngY_0)$ ，黑色区域为障碍物，点A的经纬度坐标为 $(latX_A, lngY_A)$ ，当无人船行驶时，无人船与障碍物的相对位置关系如下：

$$\begin{aligned}
 OA &= OD + DG \\
 OC &= OA \times \cos \Psi \\
 OB &= OA \times \sin \Psi \\
 AD &= \text{Length} \\
 OD &= W_s / 2
 \end{aligned} \tag{2.2}$$

$$\begin{cases} \text{lat}X_A = \text{lat}X_O - (\text{Length} + W_s / 2) \times \cos \Psi \\ \text{lng}Y_A = \text{lng}Y_O + (\text{Length} + W_s / 2) \times \sin \Psi \end{cases} \tag{2.3}$$

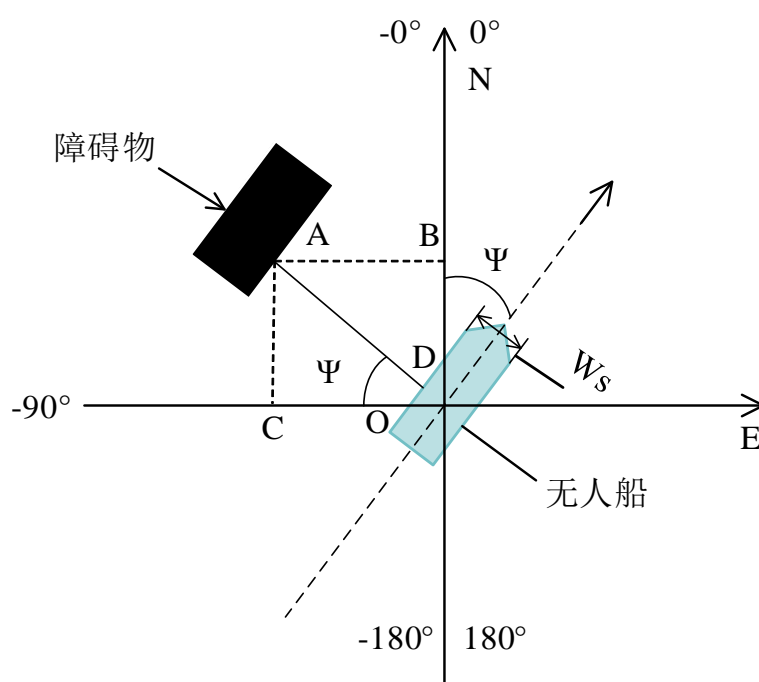


图 2.1 无人船运动模型

Fig 2.1 Model of unmanned ship movement

2.2 LOS 制导算法

LOS 制导算法通过保持船只与目标之间的直线视线路径来引导船只，沿着与目标之间的最短路径进行移动，从而最大程度地减少船只航行角度误差和路径长度偏差。其原理是在预先设定的一段航迹上选取视线点来求解无人船在每个航迹点处的期望航向角^[24]。LOS 在制导系统中，可以用来计算目标转向角，作为航向控制系统的输入^[25]。LOS 航迹跟踪原理如图 2.2 所示。

Fig 2.2 LOS guidance Strategy principle

在 LOS 制导算法中, 由操作舵控制的无人船会根据航向角 φ 与 LOS 角 φ_{LOS} 之间的偏差来跟踪参考航迹, 而参考点 $P_{LOS}(x_{LOS}, y_{LOS})$ 可以通过下列表达式求出:

$$\begin{cases} (x_{\text{LOS}} - x_0)^2 + (y_{\text{LOS}} - y_0)^2 = R_{\text{LOS}}^2 \\ \frac{y_{\text{LOS}} - y_i}{x_{\text{LOS}} - x_i} = \frac{y_{i+1} - y_i}{x_{i+1} - x_i} \end{cases} \quad (2.4)$$

$$\|O_0 - P_{i+1}\| \leq R_{i+1} \quad (2.5)$$

2.3 基于栅格法的无人船环境建模

构建一个适当的环境模型是实现无人船路径规划的基础任务。无人船的路径规划算法基于环境建模，需要捕获环境信息并构建代表无人船在行驶过程中的真实环境模型^[26]。无人船环境建模需要采用特定的数学模型来表示无人船的工作环境，这项技术一直是无人船研究中的一个关键领域和热点问题。建立合理的环境模型有助于降低路径规划的搜索路径长度，提高路径规划的效率^[27]。无人船环境建模不仅要根据实际航行环境和算法应用场景进行区分设计选择，其中已知水域内的环境可以通过 ECDIS 等多种方式进行建模，而面对未知水域或者水下未知障碍物时需要利用无人船环境感知技术进行探测避障，并创建相应环境模型。

栅格法由于其建模简单，适用范围广，易于修改，能够直观地表示水域和障碍物等环境特征。栅格地图可以有效地用于碰撞检测和避障，通过检查无人船预计路径上的栅格单元可以有效的来预测潜在的障碍物。还可根据具体应用场景选择合适的栅格粒度大小，可以通过动态更新栅格值来实时提供环境信息，适应工作环境发生的变化。所以栅格法环境建模更适合应用本研究对无人船路径规划算法的环境建模。

2.3.1 栅格法环境建模

建立栅格地图模型是采用一定的数学建模方法来表示无人船行驶的环境区域，将无人船行驶的环境区域划分为一系列大小相同的栅格。通常栅格形状为正方形或矩形，在本实验中将无人船规划的实验测试环境划分为一系列大小相同的正方形栅格空间。栅格尺寸的大小与行驶区域中的障碍物的复杂程度有关，根据障碍物大小和数量及无人船行驶区域大小来确定栅格尺寸，这是建立栅格地图的第一步。栅格环境模型是事先通过无人船上的传感器把障碍区域的大小按照比例映射在栅格地图中，将无人船行驶环境转变成一个二维的静态空间，并根据该栅格区域是否可通行将栅格地图中的栅格分为不可通行区域和可通行区域。通常无人船不可通行区域用黑色栅格表示，而可通行区域用白色栅格来表示，根据可通行区域与不可通行区域的情况将栅格赋予不同值。如果障碍物所占的区域在一个栅格中未占满，则视为一个栅格。

本文采用栅格法环境建模，用 MATLAB R2020B 进行环境建模的仿真实验，为了客观的表示栅格法环境建模，作如下定义：

定义 1：对设置好的栅格地图设置栅格序号 $G(i, j)$ ， i 为栅格地图的行数 ($i=1, 2, 3, \dots, n$)， j 为栅格地图的列数 ($j=1, 2, 3, \dots, n$)，

定义 2: 可达区域 $S_\phi = \{i, j | ocup(i, j) = 1\} \subseteq S$, 其中 $ocup(i, j)$ 为栅格值。如式(2.6)所示:

$$ocup(i, j) = \begin{cases} 1, \text{无障碍物} \\ 0, \text{有障碍物} \end{cases} \quad (2.6)$$

定义 3: 不可达区域 $S_\Delta = \{i, j | ocup(i, j) = 0\} \subseteq S$; $S_\phi \cap S_\Delta = 0$ 且 $S_\phi \cup S_\Delta = 1$;

定义 4: 无人船开始在栅格地图中规划路径时, 通过在节点对象中添加一个属性来标记该栅格是否已经被访问过。在节点对象中添加一个名为 **visited** 的属性, 如果遍历完一个栅格可以把对应节点的 **visited** 属性设为 **true**, 表示该节点已经被访问过。

定义 5: 栅格地图所采用的坐标系为笛卡尔直角坐标系。二维标识坐标与 MATLAB 中矩阵轴模式编号相同, 用一组行列编号栅格。笛卡尔直角坐标系中 x 轴正方向和 y 轴正方向对应正东、正北方向。二维标识与直角坐标系具体转换关系如下:

$$x = (j - 0.5) * len \quad (2.7)$$

$$y = (n - i + 0.5) * len \quad (2.8)$$

$$i = n - y / len + 0.5 \quad (2.9)$$

$$j = x / len + 0.5 \quad (2.10)$$

式中, n 为栅格矩阵 G 的维度, i, j 为二维标识坐标系下的栅格行列对, len 为海图与栅格地图之间的边长缩放比, mod 为取余函数, ceil 为取整函数。

通过 MATLAB R2020B 首先设置一个 30×30 的栅格地图, 地图中将障碍物设置为黑色区域, 无人船不可以通过, 将无人船可通行的区域设置为白色, 设起始点和终点颜色为红色区域和蓝色区域, 其中障碍物是无规则随机分布的, 如图 2.3 所示。

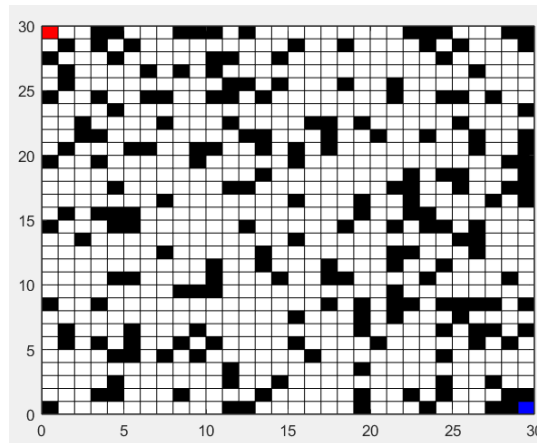


图 2.3 栅格环境建模仿真图

Fig 2.3 Grid environment modeling and simulation diagram

2.4 海图栅格法环境建模

2.4.1 海图二值化处理

在对海图进行栅格法环境建模时，需要考虑到设计栅格的大小比例，使其栅格地图的分辨率合适，一个合适的栅格不仅可以清晰的表示航行环境中的可行区域和障碍区域，还能保障无人船在栅格地图环境里的路径规划算法运算效率。栅格的大小需要根据无人船的实际工作区域来划分，在卫星地图或者电子海图上确定无人船的工作区域，然后根据截取的相关地图的比例尺确定栅格化后的海图的比例尺的大小及长宽比例，以此为依据选择适当的栅格地图。由于不同的航行区域里面障碍物的数量和大小也不同，需要的栅格大小也不同，所以当用 MATLAB R2020B 对所截取的电子海图或者卫星地图进行读取后，根据航行要求选择适当的栅格大小，栅格大小的选取影响着算法效率。在进行确定海图栅格大小时，先根据栅格地图的面积确定地图的长宽比例，以此为依据进行对地图的长宽比例进行设计，使其与原海图的长宽比例相同。然后对其进行二值化处理，使二值化后的海图只有黑白两种颜色。其中，黑色区域代表障碍物区域，白色区域代表可通行区域，在进行编程时，可以根据需要调整代码的阈值，选择合适的对比度。

如下图所示，先对图 2.4 进行海图栅格化中的二值化处理（该海图数据来自中国海事局 ECDIS 海图）。

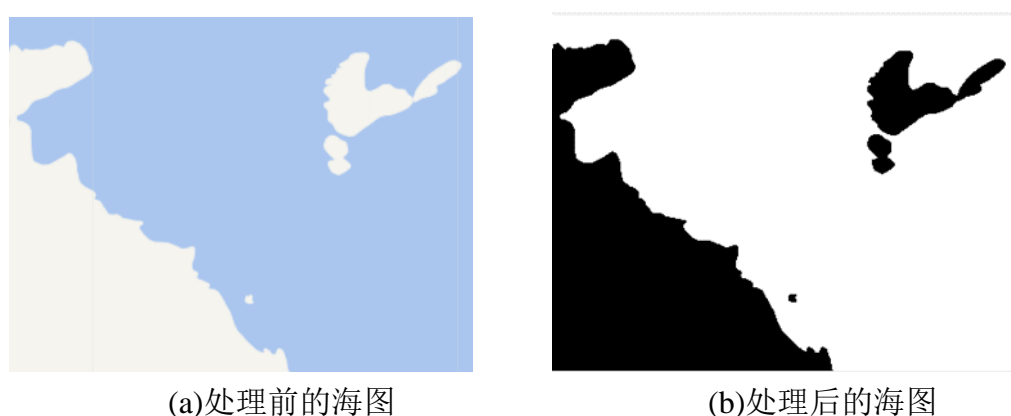


图 2.4 海图二值化处理

Fig 2.4 Binary processing of nautical charts

2.4.2 海图栅格化处理

在对电子海图进行二值化处理完之后，对其中的障碍物和可行区域进行栅格化处理。对电子海图栅格法环境建模的关键是用像素化的方法把航行中的障碍物和通行区域合理表示。处理后的环境像素由一个个障碍物栅格和可行区域栅格组成。对栅格进行赋值，对可行区域赋值为0值，表示无人船可以在这个区域行驶，对障碍物栅格赋值为1表示船舶无法在该区域中通行。由于在栅格化的过程中，在由一整块障碍物栅格组成的障碍物区域的边缘在栅格化的时候，由于一个栅格中会出现一部分障碍物区域一部分是可行区域这一现象。所以，统一规定，读取每一个栅格中心点的位置，如果该栅格的中心点位置为障碍物，则对这个栅格赋值为1，反之，若中心点位置为可行区域，则对这个栅格赋值为0，根据海图长宽比例设置海图栅格的组成数量。在此海图中，海图的长宽比例为3:4，所以设置栅格地图数量为90*120。将环境中障碍物未占满一个的栅格单元作为一个完整障碍物栅格单元，最大限度保障船舶航行安全。海图栅格化处理结果如下图2.5所示：

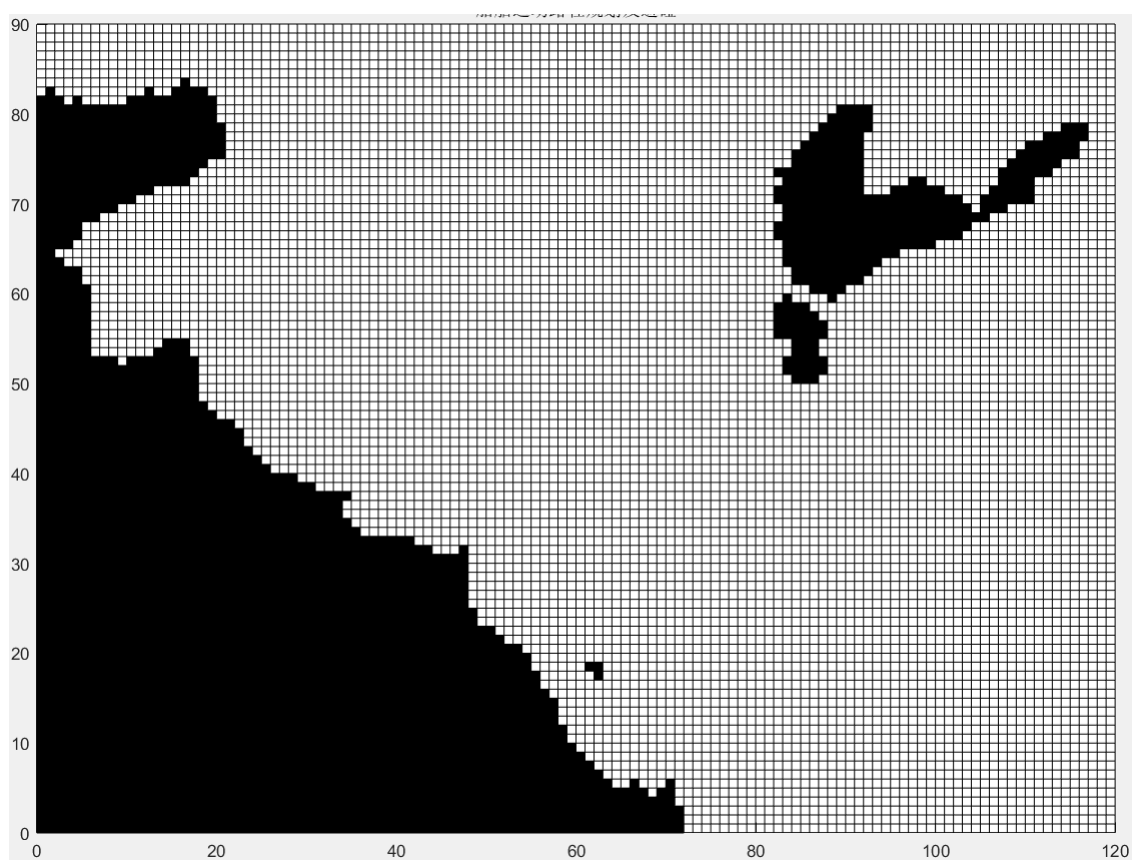


图 2.5 海图栅格化处理

Fig 2.5 Rasterization processing of nautical charts

2.5 本章小结

本章对无人船数学模型进行分析说明，并讲述了无人船进行路径跟踪的常用算法——LOS 制导算法基本原理。无人船环境建模是研究无人船路径规划算法的前提，本章详细介绍了无人船环境建模中的栅格环境建模的基本原理与建模流程，首先通过读取电子海图的数据选择合适的栅格地图的尺寸比例，然后通过对栅格地图进行二值化处理得到二值化之后的海图，对障碍物区域和可行区域进行赋值，并规定每一个栅格的中心所在区域来确定这个栅格的整体区域是属于障碍物区域还是可行区域。处理完后对二值化地图进行栅格化处理，完成了无人船执行任务区域的栅格环境建模。处理完后可以通过对比原二值化后的海图，手动调节设置障碍物栅格，使栅格地图更加准确。

3 路径规划算法概述与改进

无人船在进行路径规划的时候，通常分为两种形式的路径规划^[28]。一种是基于静态环境下的全局路径规划（Local Path Planning）：在起始点和目标点都被确定后，通过选择不同的路径规划算法来规划一条适合无人船行驶的无碰撞安全行驶路径；而另一种是当无人船行驶环境区域附近存在船舶或者周围存在动态障碍物时，如果用全局路径规划算法，则可能有碰撞的风险，所以需要用到局部路径规划算法来实时动态避障，局部路径规划时需要借助船上的传感器装置来实时提供环境信息，以保证航行安全。路径规划技术作为无人船航行研究中的关键技术，如果只是靠一个路径规划算法往往是不足以满足航行需求的，所以在实际应用中，无人船路径规划时通常会将全局路径规划算法与局部路径规划算法相结合。因为无人船航行场景和航行需求的不同，路径规划算法也会结合规划目标来做出改变^[29]。

根据无人船对环境信息获取的方式不同，将路径规划算法分为动态和静态^[30]。将全局路径规划在已知环境地图内进行路径规划属于静态路径规划，局部路径规划需要实时对周围环境信息进行收集属于动态路径规划。本文中的全局路径规划研究都属于静态路径规划。

3.1 全局路径规划

全局路径规划算法的研究现在已经较为成熟，在船舶智能化研究领域有着重要地位。要想实现全局路径规划的前提是建立一个无人船航行环境地图模型，在该环境地图中进行路径规划。

全局路径规划算法的优势在于整个环境地图内的可行区域和非可行区域区别明显，容易找到最优路径，且无人船对实时性的要求相对较低。适用于在整个环境地图模型内进行路径搜索。但是由于需要对整个环境地图信息进行储存，所以计算量较大。当对于无人船处于动态环境或者行驶在大量障碍物的复杂环境地图中，需要环境信息的精确信息较高。而且还可能无法实时响应并生成最优路径，无法处理算法规划过程中出现的临时障碍物。对于某些全局路径规划算法，当环境地图较为复杂时，可能最后搜索的最优路径并不是一条最终确定路径。目前较为代表的几个全局路径规划算法有：A*算法、Dijkstra^[31]算法和蚁群算法。其中 A*算法和 Dijkstra 算法都是基于图形法，而蚁群算法属于进化算法^[32]。

3.1.1 A*算法

A*算法是由 Peter Hart 等人^[33]在 1968 年研究如何利用计算机科学技术来实现有效的路径规划过程中，首次开发并提出的一种基于 Dijkstra 算法^[34]和最佳优先算法 (BFS)^[35]的启发式算法。A*算法与其他算法相比如贪心算法^[36]或 Dijkstra 算法能在找到

最短路径的同时尽量减少算法的计算量，提高了搜索效率，减少了路径节点等优点^[37]，广泛应用于多个领域。

(1):传统 A*算法代价评估函数

A*算法之所以在众多算法中脱颖而出，得益于其算法中启发式函数的设计。通过设定适合的启发函数，使其搜索方向朝着最有可能到达目标的方向进行，直至到达目标点。这一设计不仅考虑到了起始节点到当前节点的实际代价成本，还预估了从当前节点到目标节点的所需要的代价成本，评估函数的设计平衡了路径的最优性和搜索的效率。A*算法的代价评估函数为：

$$f(r_i) = g(r_i) + h(r_i) \quad (3.1)$$

式(3.1)中： $g(r_i)$ 表示为从起始节点到当前节点 r_i 的实际代价成本； $h(r_i)$ 为从当前节点到目标节点 r_i 的最优路径的估计代价成本； $f(r_i)$ 表示为从起始节点出发经由当前节点 r_i 再到目标节点的估计代价成本。式(3.1)中的 $g(r_i)$ 与 $h(r_i)$ 存在相互制约的关系，当估计代价成本 $h(r_i)$ 不为0时，实际代价成本 $g(r_i)$ 越大，则表示算法搜索的节点增多，搜索效率下降，反之，当实际代价成本 $g(r_i)$ 减小时，则从当前节点到达目标节点的最优路径估计成本 $h(r_i)$ 增大，算法搜索节点变少，搜索效率上升。当估计代价成本 $h(r_i)$ 为0时，即： $f(r_i) = g(r_i)$ ，此时A*算法变为Dijkstra算法。

A*算法示意图如图3.1所示，图中绿色圆圈为目标点，蓝色三角形为起始点，黑色区域为障碍物，蓝色路径为A*算法搜索的路径，白色区域是A*算法可搜索区域，灰色区域是A*算法已经搜索过的区域。

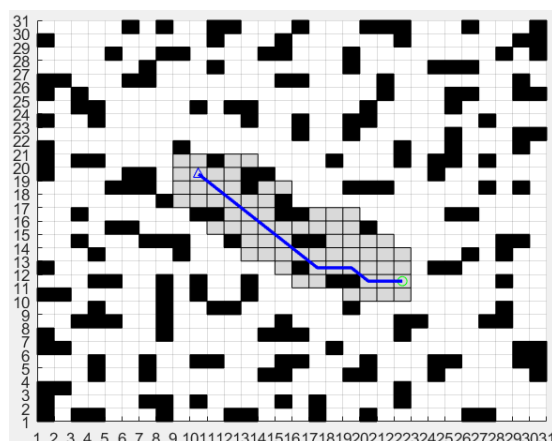


图 3.1 A*算法示意图

Fig 3.1 Schematic diagram of A* algorithm

(2):传统 A*算法寻路过程

首先进行预处理，将栅格地图中每个栅格中心位置设置为父节点，每个父节点周围相邻着八个子节点。①：每个父节点都储存着节点信息，建立开启链表 *OpenList* 和关闭链表 *CloseList*，定义起始节点和目标节点。②：将起始节点定为父节点并存入关闭链表中，将所有的子节点存入开启链表中。③：起始节点按照评估函数找寻 *OpenList* 中 $f(r_i)$ 值最小的节点 a 并移到此处。把节点 a 当作当前节点。若节点 a 为目标节点，程序结束。输出最优路径距离，如果不是目标节点，则继续循环。④：对当前节点 a 进行判断，周围邻域子节点是否在链表中，如果在，则忽略此节点，如果不在，则添加进 *OpenList* 中。继续按评估函数找寻周围处于 *OpenList* 中的相邻节点，如果该相邻节点的 $f(r_i)$ 值小于原起始节点到节点 a 的 $f(r_i)$ 值。则用新的 $f(r_i)$ 值代替旧的 $f(r_i)$ 值，把相邻节点的父节点改为当前节点 a。重复执行步骤③、④，直至当前节点是目标节点或者是 *OpenList* 中没有存储节点为止。

传统 A*算法在栅格地图上运行时，其搜索路径上两个相邻节点之间的距离是按照两节点中心点的欧几里得距离作为路径长度的估计代价成本：

$$h(x_i) = \sqrt{(x_E - x_s)^2 + (y_E - y_s)^2} \quad (3.2)$$

上式中， (x_E, y_E) 是目标节点的坐标； (x_s, y_s) 是当前节点 a 的坐标。传统 A* 算法的流程图如下所示。 $d(X, Y)$ 为节点 X 到节点 Y 的距离。

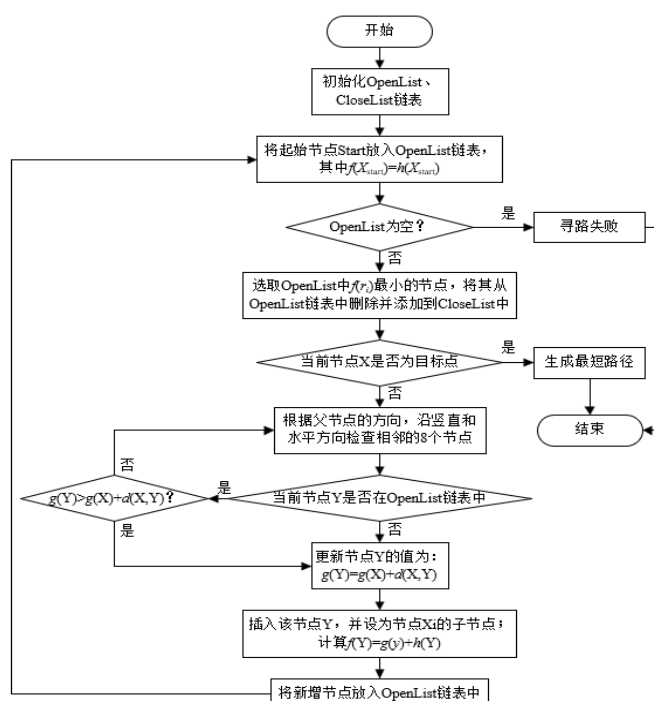


图 3.2 A*算法流程图

Fig 3.2 Flowchart of A* algorithm

A*算法的缺点是在存储开放列表和封闭列表时, 可能需要占用大量的内存资源, 尤其在搜索空间非常大的情况下; A*算法规划的路径转向点较多, 路径不够平滑; 由于A*算法对启发式函数的依赖较大, 如果没有合适的设计启发函数, 可能会导致A*算法无法找到最优路径或导致搜索效率降低。

3.1.2 蚁群算法

蚁群算法^[38] (Ant colony optimization)是一种模拟蚂蚁觅食的用来寻找路径仿生智能优化算法, 在1990年代被M.Dorigo提出。算法的原理是来源于蚂蚁在找寻食物的过程中会在行驶过后的路径上留下自己的信息素, 通过信息激素传递信息进行交流, 当某只蚂蚁在某地发现食物或发现重要信息时, 会重复多次释放信息激素导致该地的信息激素浓度上升^[39]。同时, 当其他蚂蚁通过这条路径走到这个地方时也会释放信息激素再次强化, 蚁群之间传递信息的激素会引导蚂蚁个体找到路径, 同时这种路径信息会受到蚂蚁个体的重复强化^[40]。当后来的蚂蚁经过这个路口时, 会选择信息素浓度较高的路径, 最优路径上的信息素浓度越来越大, 最终蚁群找到最优路径。

蚁群算法能自动调整搜索策略，适应复杂多变的环境。作为一种启发式算法，能在全局范围内进行搜索，有效避免了局部最优解^[41]；蚁群算法大量应用于路径规划，车辆调度，货物分配等领域^[42]。其算法流程如下。

第一步：初始化，设置蚁群的数量规模，迭代次数，信息素挥发系数，信息素重要性系数，距离重要性系数，启发函数的影响程度，在开始时假设所有路径上的信息素浓度相同。蚂蚁进行节点选择根据每个节点之间的距离和路径上的信息素差异来选择下一个节点。蚂蚁从节点*i*到节点*j*的状态转移方程如式(2.8)所示：

$$p_{ij}^k = \begin{cases} \frac{[\tau_{ij}]^\alpha [\eta_{ij}]^\beta}{\sum_{k \in \text{allowed}_k} [\tau_{ij}]^\alpha [\eta_{ij}]^\beta}, j \in \text{allowed}_k \\ 0, \text{otherwise} \end{cases} \quad (3.3)$$

$$\eta_{ij} = \frac{1}{d_{ij}} \quad (3.4)$$

其中， α 是信息素启发因子，表示路径搜索过程中信息素浓度对蚂蚁搜索的重要性； β 是期望启发因子，表示路径搜索过程中节点间可见度的相对重要性。 $\text{allowed} = \{0, 1, \dots, n-1\}$ 表示蚂蚁*k*可以选择的节点； τ_{ij} 表示蚂蚁搜索路径上的信息素浓度； η_{ij} 表示从节点*i*到节点*j*的期望值； d_{ij} 表示节点*i*到节点*j*之间的欧式距离。欧式距离的表达式为： $\left((x_1 - x_2)^2 + (y_1 - y_2)^2\right)^{1/2}$ 。

第二步：将所有蚂蚁放置起始点，每只蚂蚁独立地根据概率选择下一个节点，此概率由当前节点到下一节点的信息素浓度和启发式信息（如两节点之间的距离）共同决定。并将每只蚂蚁走过的路径节点信息记录在禁忌列表中，排除非法以及障碍物节点，如果没有合法节点，则终止节点的探索。

第三步：计算每只蚂蚁走过的路径长度，向路径上的每段添加信息素，路径越短，添加的信息素越多，记录当前迭代次数中的最优路径，同时对每条路上的信息素进行更新。检查是否到达最大迭代次数或其他停止条件，若到达最大迭代次数则找出所有迭代次数中的最优路径继续下一步输出结果，若未达到终止条件，则重复第二步和第三步继续迭代。

蚁群算法的具体流程图如图 3.3 所示。

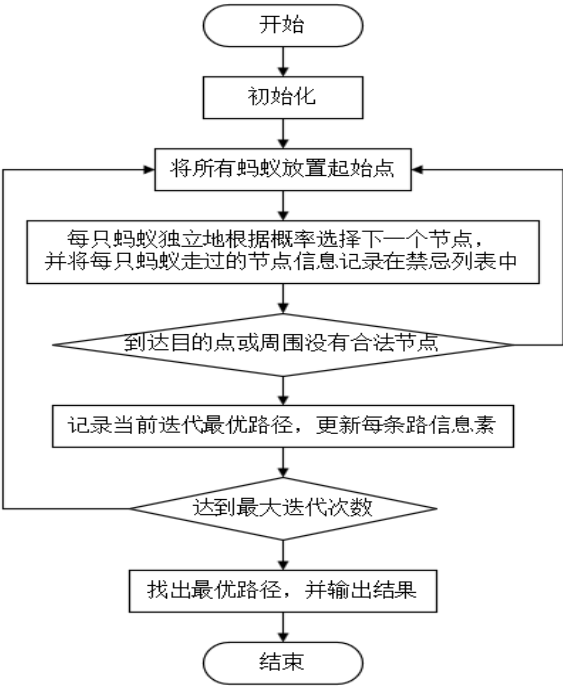


图 3.3 蚁群算法具体流程图

Fig 3.3 Specific flowchart of ant colony algorithm

3.1.3 路径搜索算法对比

现在通过对蚁群算法和 A*算法进行路径规划实验，实验所用的栅格地图全部相同，起始点和终止点位置相同，对比两种典型路径搜索算法的性能。蚁群算法搜索路径示意图如图 3.4(a)所示。A*算法搜索路径示意图如图 3.4(b)所示。图中蓝色圆点代表起始点，黑色区域为障碍物，灰色代表已经搜索区域，绿色区域代表可行区域，红色圆点代表目的点。

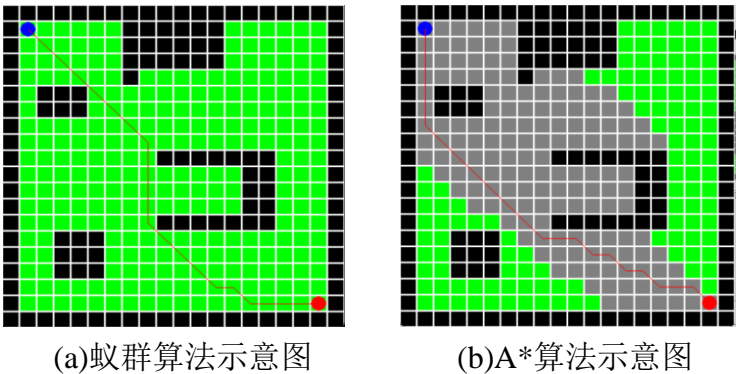


图 3.4 两种路径搜索算法示意图

Fig 3.4 Schematic diagram of two path search algorithms

表 3.1 两种路径搜索算法评价

Tab 3.1 Evaluation of Two Path Search Algorithms

路径搜索算法	路径长度	转弯次数	运行时间
蚁群算法	23	8	2.854
A*算法	24	9	0.015

由表 3.1 中可知,相比于蚁群算法,A*算法路径长度为 24,转弯角度在(0° , 90°)的次数为 9,在这两个参数上面,蚁群算法和 A*算法相差较小,但在运行时间上,A*算法远比蚁群算法路径搜索速度快的多。由于 A*算法在栅格地图中的收敛速度远大于蚁群算法,且 A*算法由于其简单易实现的特点被广泛使用,本文采取 A*算法作为全局路径规划算法。

3.2 局部路径规划

局部路径规划需要根据传感器所收集的环境信息对周围环境地图进行更新,在规划无人船运动方向是按朝着向目标点靠近的同时还需要对航行过程中突然出现的动态障碍物进行一个实时避障。通过不断跟新航行环境数据和路径规划结果来持续的调整航向和航速,常见的局部路径规划算法有 RRT 算法,动态窗口法等,本文对传统动态窗口法进行改进,并将改进后的动态窗口法作为无人船局部路径规划算法。

3.2.1 RRT 算法

快速扩展随机树(Rapidly-exploring Random Tree, RRT)算法最早由 Lavalley 等人在 1998 年提出,RRT 算法基于随机取样的原理,通过构建一颗随机树来探索状态空间。以自己的初始点作为根节点^[43],通过在工作空间里随机添加一个又一个叶子节点,每次迭代出一个位姿点,并寻找与之最近的节点作为父节点,然后在父节点与随机状态之间按照某种策略(如判断是否与障碍物碰撞、检查新扩展的节点是否满足接近终点的条件、或者是否能直接连接到终点)扩展出新的节点。直到叶子节点到达目标节点或者达到目标节点的一定判断范围内时结束搜索,最后可以从根节点生成一条连通目标点的全局路径。

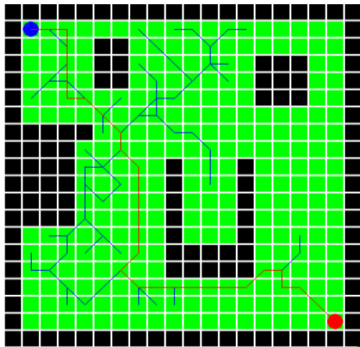


图 3.5 RRT 算法搜索示意图

Fig 3.5 Schematic diagram of RRT algorithm search

RRT 算法是一种有效的路径规划算法，尤其适用解决高维空间和复杂约束条件下的路径规划问题。RRT 算法因其随机采样算法的特性对三维空间和更高维度空间的路径规划具有出色的效果^[44]，因此被广泛应用于各种路径规划问题中。RRT 算法的伪代码如表 3.2 所示。

表 3.2 RRT 算法伪代码

Tab 3.2 algorithm pseudocode for Rapidly-exploring Random Tree

RRT 算法伪代码：
1:class Node(object); 2:def __init__(self, x, y): 3:def random_node(self): 4:def get_nearest_list_index(node_list, rnd): 5:collision_check(new_node, obstacle_list); 6:planning(self): 7:if not self.collision_check(new_node, self.obstacleList); 8:min_index = self.get_nearest_list_index(self.nodeList, rnd): 9:nearest_node = self.nodeList[min_index]: 10: continue; 11: self.nodeList.append(new_node): 12: if d <= self.expandDis: 13: print("Goal!!") 14: break

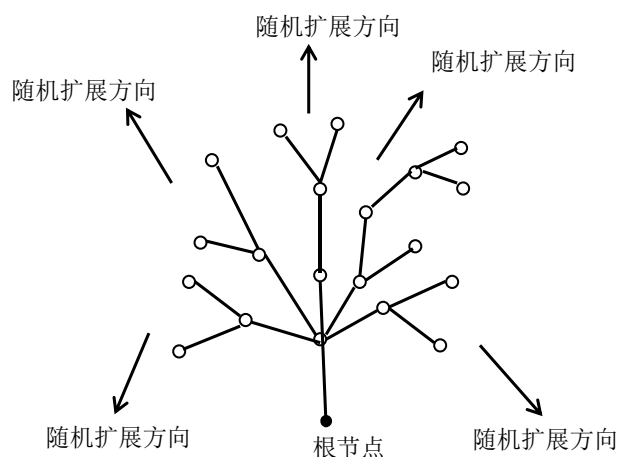


图 3.6 RRT 算法节点扩展方式

Fig 3.6 RRT algorithm node expansion method

3.2.2 DWA 算法

动态窗口法（Dynamic Window Approach, DWA）是一种常用的局部路径规划算法，与其他大部分局部路径规划算法不同的是，传统 DWA 算法充分考虑到了无人船航行过程中自身的机械特性限制和环境对船速的约束^[45]，DWA 算法生成的轨迹更符合无人船实际航行轨迹^[46]。DWA 算法最早由 FOX 等人^[47]在 1997 年提出，为了寻找在最短时间内到达目标点的一条无碰撞路径。通过在船舶线速度 v 的取值范围内和角速度 ω 的取值范围内进行采样一个模拟航行周期内的多种速度组合 (v, ω) ，根据轨迹评价函数计算出每组速度组合的轨迹评价值，从中选择最优轨迹对应的线速度和角速度，将该速度组合作为当前无人船运动的速度指令，不断循环上述步骤直至到达目标点。

DWA 算法采样的速度矢量应该考虑到无人船在航行过程中可能达到的线速度和在航行过程中对可能造成碰撞风险的动态障碍物避免碰撞的限制速度。

（1）运动速度约束

无人船自身速度最值范围限制为：

$$V_s = \{(v, \omega), v \in [v_{\min}, v_{\max}] \wedge \omega \in [\omega_{\min}, \omega_{\max}]\} \quad (3.5)$$

式(3.5)中， v_{\min} ， v_{\max} 分别为无人船的最小线速度和最大线速度， ω_{\min} ， ω_{\max} 分别为无人船的最小角速度和最大角速度。

（1）安全性约束

为了确保动态窗口法在避障过程中的安全性，对无人船的采样速度应该能够使得无人船在接触障碍物之前时停止运动。因此，采样速度的选择需要考虑无人船的制导

距离的限制。所以，在一个模拟航行周期内采样速度矢量在最大减速度制动下，能够使得无人船在离障碍物的距离处于最小安全距离外停下来的速度集合 V_a 应满足^[48]：

$$V_a = \left\{ (v, \omega) \mid v \leq \sqrt{2 \cdot \text{dist}(v, \omega) \cdot \dot{v}_b} \wedge \omega \leq \sqrt{2 \cdot \text{dist}(v, \omega) \cdot \dot{\omega}_b} \right\} \quad (3.6)$$

式(3.6)中， $\text{dist}(v, \omega)$ 为船舶以速度矢量为 (v, ω) 时对应轨迹上与障碍物的最近安全距离。传统 DWA 算法路径规划示意图如图 3.7 所示，角速度，线速度，无人船姿态（弧度）等信息如图 3.8 所示，起始位置为 $(3.5, 3.5)$ ，目标位置为 $(18.5, 11.5)$ 。

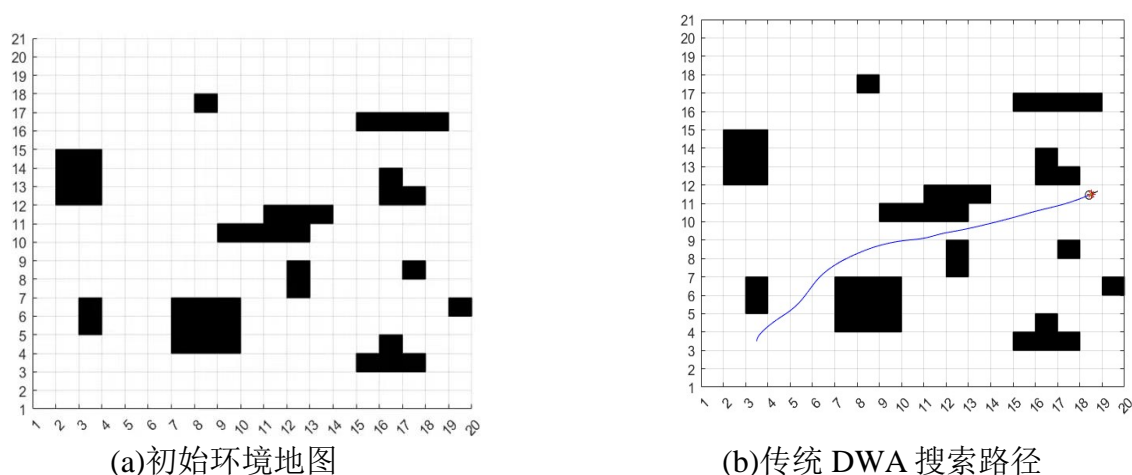


图 3.7 传统 DWA 算法搜索路径示意图

Fig 3.7 Schematic diagram of traditional DWA algorithm search path

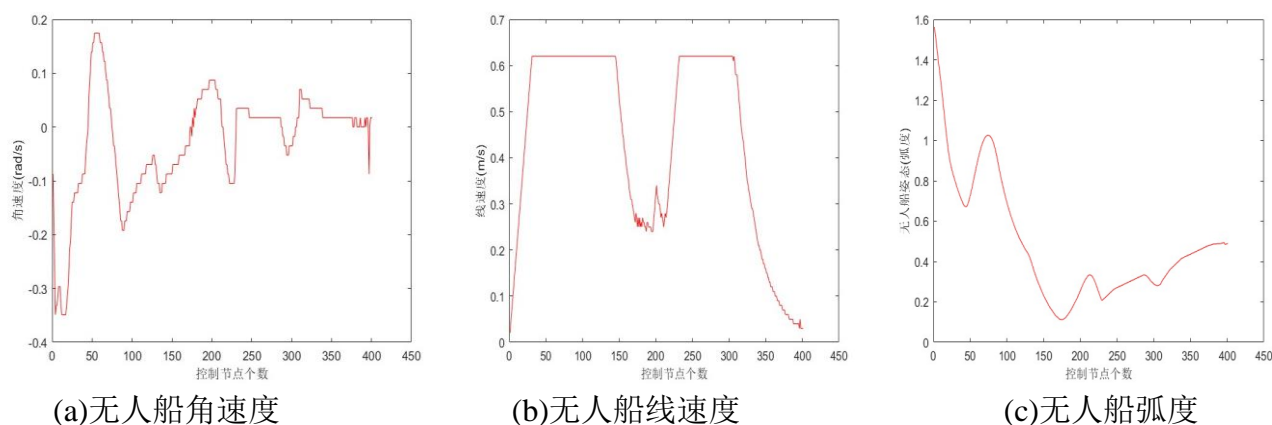


图 3.8 传统 DWA 算法搜索路径数据图

Fig 3.8 Schematic diagram of traditional DWA algorithm for searching path data

3.3 路径规划算法改进

3.3.1 改进 A*算法及算法实例

传统的 A*算法在栅格地图环境里从起始点到目标点的搜索路径经常会出现转折点较多,且很多转折点不符合无人船在真实环境中的运动情况。无人船如果按照传统 A*算法规划的路径进行实际路径规划的话,无人船所占的栅格中心往往会与栅格障碍物的顶点连线成相切关系,导致搜索路径不够平滑、存在发生碰撞的风险、且规划的路径通常不满足无人船的最优路径需求。在存储开发链表和封闭链表时会造成过多数据占用内存资源的情况,降低了路径搜索效率。针对上述传统 A*算法的不足,并根据无人船行驶的实际情况,对传统 A*算法提出无人船任意方向转向改进,以及其他一些优化处理。

无人船在水中以自己为圆心,可以朝任意方向转动,路径规划算法不仅要尽量在规划好的栅格地图中找到最优路径,而且所规划的路径要尽可能考虑到无人船实际运动情况。改进 A*算法的过程如下。

(1)传统的 A*通过 *BestList* 链表来存放所有得到的节点,而改进后的 A*算法优化了节点的数据结构,使用类来表示节点,可以更方便地管理节点信息,包括了父节点、位置、 $g(r_i)$ 的值、 $h(r_i)$ 值和 $f(r_i)$ 值;

(2)无人船当前的起点 S,利用水波纹算法,从中心向四周搜索栅格地图 MAP 中还未遍历的栅格 E,即栅格值为 $ocpu(x,y)=1$,将 E 作为 A*算法终点。同时定义了两个辅助函数: *get_neighbor* 函数来获取当前节点 S 的相邻节点列表, *cost* 函数用于计算从起点到目标点的估计代价成本;

(3)改进 A*算法使用了曼哈顿距离作为启发式函数,用于估计当前节点到目标节点的距离,算法搜索并筛选出最短路径对应的节点,放入 *BestList* 链表中。改进 A*算法通过一些函数来辅助计算点与点之间的距离和某个节点到之前搜索路径的直线距离, *PointDistance* 函数和 *StraightLineDistance* 函数。 *pointCompare* 函数来对比其最短路径对应节点的坐标,以确定节点的坐标,并按顺序排列;

(4)把最短路径所对应的坐标点按坐标顺序依次与起始点用线段相连接,这些节点坐标的选取应该与初始节点的横坐标或者纵坐标不相等,且起始点与该坐标点相连的线段最长,障碍物的栅格顶点到该线段的最短垂直距离均应该大于一个栅格,满足路径的安全性;

改进后的 A*算法与传统 A*算法的对比图如下：灰色区域为算法搜索过的区域，绿色区域代表可行区域。黑色区域代表障碍物，算法的起始点是蓝色圆点，终点为红色圆点。

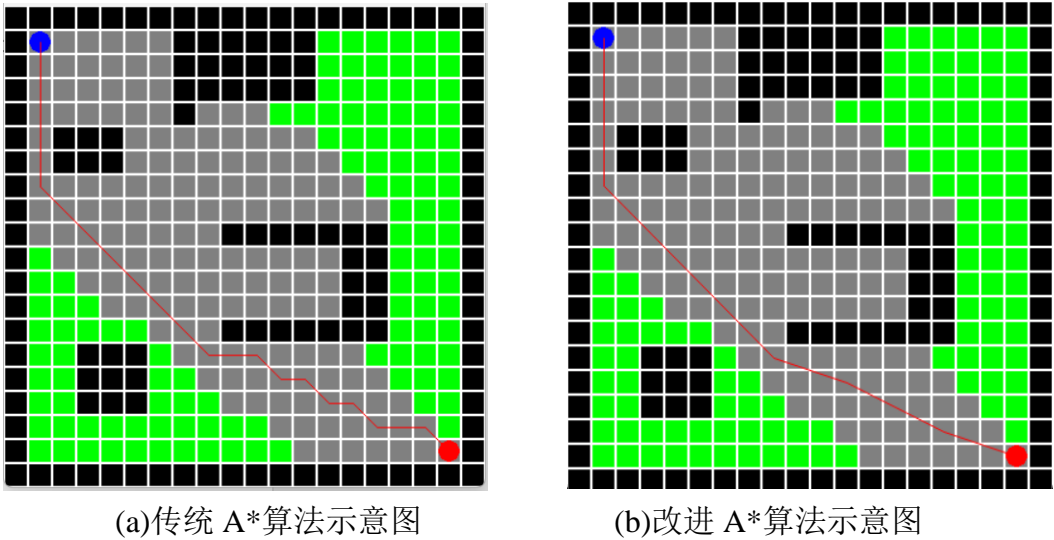


图 3.9 两种算法的路径示意图

Fig 3.9 Path diagrams of two algorithms

表 3.3 传统 A*算法与改进后的 A*算法数据分析

Tab 3.3 Data Analysis of Traditional A* Algorithm and Improved A* Algorithm

两种算法	节点数	路径长度	算法运行时间
传统 A*算法	9	24	0.012
改进后的 A*算法	2	13	0.018

图 3.9 传统 A*算法与改进后的 A*算法相比可知，改进后的 A*算法转折点更少，路径更加平滑。从数据对比表 3.3 可得，改进后的 A*算法在路径长度上比传统 A*算法更加简短，在实际无人船航行中可以有效降低能量损耗。通过这些改进，改进后的 A*算法能够更高效地搜索最优路径，并且在处理大规模地图时具有更好的性能。

3.3.2 改进 DWA 算法

传统的 DWA 算法和其拓展算法在面对不同的障碍物分布时大多数采取的是用恒定的权重参数，且评价函数的权重需要手动设置，但是这种设置方式可能不适用于所有场景，导致算法在某些场景中规划的路径效果并不是很理想。因此，本章提出了一种改进 DWA 算法，通过引入模糊控制方法^[49]，使 DWA 算法与模糊控制相结合，通过分

析实时环境信息与无人船的运动状态来调整评价因子的权重，以适应不同的场景和任务需求。

模糊控制是一种基于模糊逻辑的控制方法，它可以处理模糊和确定的信息，并生成模糊的控制策略。通过模糊控制器^[50]，我们可以根据实时的环境和无人船运动状态，计算出适合当前情况的评价因子^[51]的权重。这样算法就可以根据实际情况进行动态调整，提高路径规划的性能和鲁棒性。

改进 DWA 算法搜索路径的过程如下

(1)：绘制地图并标记地图中障碍物的位置，定义起始点和目标点坐标，计算初始航向角，使无人船朝着目标点以正确的方向前进，如果无人船的初始位置与目标点重合，则航向角为 π 。

(2)：创建无人船的初始状态，无人船的初始状态包括位置、航向角、线速度、角速度。并设置无人船的仿真参数，包括无人船每次的仿真步长和前向模拟时间。

(3)：定义无人船的碰撞半径，用于检测是否与障碍物发生碰撞。设置无人船运动学限制，包含了无人船的最高线速度、最高角速度、最大线加速度和角加速度、线速度分辨率和角速度分辨率。并初始化了评价函数参数，设置规划路径的最大步长，防止无限循环。

(4)：调用路径规划函数 DWA，并传入相关参数的信息，绘制规划路径。评价函数中的权重因子用 α_1 和 γ_1 表示自适应变化。

改进 DWA 算法路径示意图如下图：所示，权重因子自适应变化图如图：所示。

蓝色三角形代表起始点，红色圆点代表目标点，蓝色圆点代表无人船，绿色线条是无人船按改进 DWA 行驶的路径。该地图的尺寸为 20*20，黑色区域代表障碍物区域。起始点和目标点的坐标分别为：(1.5, 2.5)、(18.5, 11.5)。

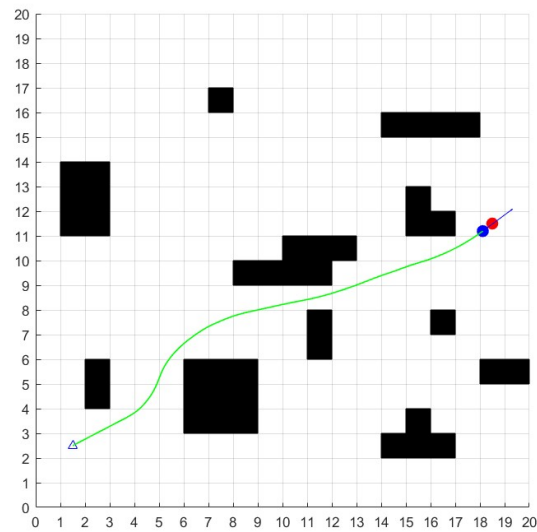


图 3.10 改进 DWA 算法的路径示意图

Fig 3.10 Path diagram for improving DWA algorithm

权重因子 α_1 和 γ_1 自适应变化图如下图所示

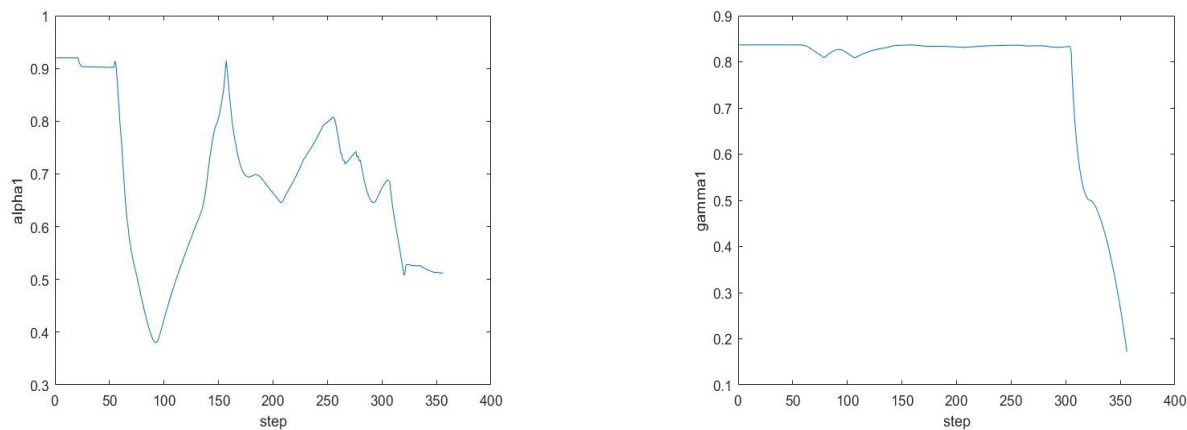


图 3.11 权重因子自适应变化示意图

Fig 3.11 Schematic diagram of adaptive changes in weight factors

3.4 本章小结

本章对无人船的路径规划算法进行了概述，分别叙述了无人船全局路径规划算法和局部路径规划算法各自适应的场景和各自的优缺点。无人船全局路径规划通过对比蚁群算法和传统 A* 算法，体现出 A* 算法适应性强，搜索速度快等优点，故本文全局路

径规划采用 A* 算法。同时，针对传统 A* 算法存在的一些缺点：搜索路径经常出现转折点较多，且不符合无人船的真实运动情况，搜索路径不够平滑，规划路径与障碍物距离过近存在危险等情况，所以对传统 A* 算法进行改进，提出了任意方向的转向改进和其他一些优化处理。对于局部路径规划算法，本章举例了 RRT 算法和 DWA 算法，由于 DWA 算法更能考虑无人船的航行过程中的自身运动特性和环境对无人船的一些约束，DWA 算法生成的轨迹更符合无人船的实际航行轨迹。所以本文局部路径规划算法采用 DWA 算法，但是由于传统 DWA 算法需要人为手动操作设置评价函数权重等参数，导致算法可能在某些地图中的路径规划结果并不是很理想。因此本章提出了一种改进 DWA 算法，通过引入模糊控制方法，使 DWA 算法与模糊控制相结合，通过分析实时环境信息与无人船的运动状态来调整评价因子的权重，可以根据实际的情况进行动态调整，提高路径规划的性能和鲁棒性，以适应不同的场景和任务需求。

4 无人船完全遍历路径规划

无人船作为水域交通系统中的重要组成部分，其路径规划技术是无人船众多技术中的核心技术，同时路径规划技术也一直是学术界和工业界的关注焦点。路径规划的有效性直接影响着无人船的任务执行能力和安全性。随着无人船技术的不断发展和应用场景的不断拓展，市场上也涌现了各种不同类型的无人船产品。其中一个评判产品性能的重要指标是产品能否实现完全遍历，本章将对无人船完全遍历路径规划进行深入研究。

4.1 路径规划的核心问题

路径规划是令无人船从起始点搜索出一条路径至目标点，且搜索过程要保障无人船的航行安全，同时还要保证路径搜索效率一直是国内外学者研究的热点。路径规划的好坏反映了无人船智能化程度的高低。在搜索路径过程中往往要考虑这些问题：

- (1) 如何选取有效的路径搜索算法，寻找从起始点至终点之间最优路径；
- (2) 在搜索路径过程中避开障碍物的能力；
- (3) 怎样构建合适的环境模型，使无人船能用路径规划算法规划路径。

4.1.1 路径搜索问题概述

无人船路径搜索是指在具有障碍物的环境地图中，按照一定的约束函数规则标准，利用搜索技术进行搜索路径并判断该条路径是否可行。在搜索过程中，需要找到一条安全路径，该路径连接起始点和目标点。路径搜索问题在不同无人实验平台领域都大致相似，都是通过一定的技术搜索手段来找到安全最优路径。

4.1.2 路径规划分类

根据不同需求，路径规划可以分为两种：点到点的路径规划和完全遍历路径规划。点到点路径规划旨在寻找起点至终点的一条最佳路径；完全遍历路径规划是在满足某些性能指标评价最优的前提下寻找一条能够覆盖任务区域内所有可达点的路径^[52]。两种路径规划都是以搜索过程中无碰撞为前提。

两种路径规划之间的联系是：多个点到点路径规划按一定的次序相互组成可以构成一个完全遍历路径规划。而两种路径规划之间的区别是：点到点路径规划旨在寻找一条安全路径，而完全遍历路径规划是寻找一条能够覆盖任务区域内所有可达点的路径，而不仅仅只是寻找从起点到终点的一条路径。两种路径规划算法都是为了实现无人船的自主航行而设计的，在实际应用中，无人船点到点路径规划适用于无人船从一

个特定位置到另一个特定位置的情况，例如海上巡逻、货物运输、海上救援等。无人船完全遍历路径规划算法适用于需要无人船在任务区域内覆盖所有可达点的情况，例如海洋地形勘测、水面污染物清理等方面。

4.2 完全遍历算法性能指标及“死角”情况

完全遍历算法性能主要由以下几个指标来衡量。

4.2.1 完全遍历算法的性能指标

(1)遍历完备率：是指无人船路径规划完成后，无人船所驶过的区域是否完全遍历地图中的所有可行区域，表示结果是一个百分比，即行使过的区域面积与所有可行区域面积的比例，又称遍历覆盖率。如式(4.1)所示：代表遍历完备率、代表已通过可行区域的面积、代表地图中所有可行区域的总面积。

$$X_{HC} = S_{HC} / S_{\phi} \quad (4.1)$$

无人船完全遍历时，可行区域遍历越多，遍历完备率越高，说明算法性能越好。

(2)路径节点数：是指无人船在算法规划完路径后，不可能按直线从起始点一直行驶至目标点，在行驶过程中难免会改变运动方向，转弯角度在 $(0^{\circ}, 90^{\circ})$ 的节点定义为 $N45^{\circ}$ ，转弯角度在 $(90^{\circ}, 180^{\circ})$ 的节点定义为 $N135^{\circ}$ 。其中每改变一次运动方向就会生成一个转向节点。对无人船行驶过程中所有的转向节点数进行相加，转弯节点总转向的度数越小，行驶路径越平滑，无人船运动更平稳。

(3)路径最优性：指在环境建模空间内完全遍历可行区域后，算法规划的路径是否为最优路径，评价指标便是这条路径的总长度。在遍历完备率相同的情况下，算法规划的无人船路径总长度越小，说明该算法的路径最优性越好。

(4)时间复杂度：是指无人船利用路径规划算法执行从起始点至目标点搜索路径的时间和无人船完全遍历环境地图的时间总和。较低的时间复杂度意味着无人船所用的路径规划算法能够用较短的时间完成在环境地图内搜索可行路径和完全遍历。时间复杂度越低，算法性能越好。

(5)空间复杂度：是指路径规划算法在搜索路径时所需的内存空间量级，包括存储搜索路径过程中的数据结构和地图信息的空间数据，较低的空间复杂度意味着算法储存数据所需的内存空间资源较少。空间复杂度越小，算法利用内存资源方面越高效。

4.2.2 “死角”问题

在无人处完全遍历路径规划中，无人船需要避开栅格地图中的障碍物，并尽可能地覆盖所有可行区域。因为无人船在这个位置无法继续探索。从算法的优越性角度来

看,“死角”情景的处理是路径规划算法性能的一个重要评价指标^[53]。当陷入“死角”时,需要搜索算法搜索最近 $ocup(x,y)=0$ 的栅格,并规划一条最短路径,使得无人船可以跳出“死角”。一个好的路径规划算法应该能够识别栅格地图中的死角并避免无人船陷入“死角”,使得无人船能够在栅格地图环境中灵活移动,并尽可能遍历所有可行区域。

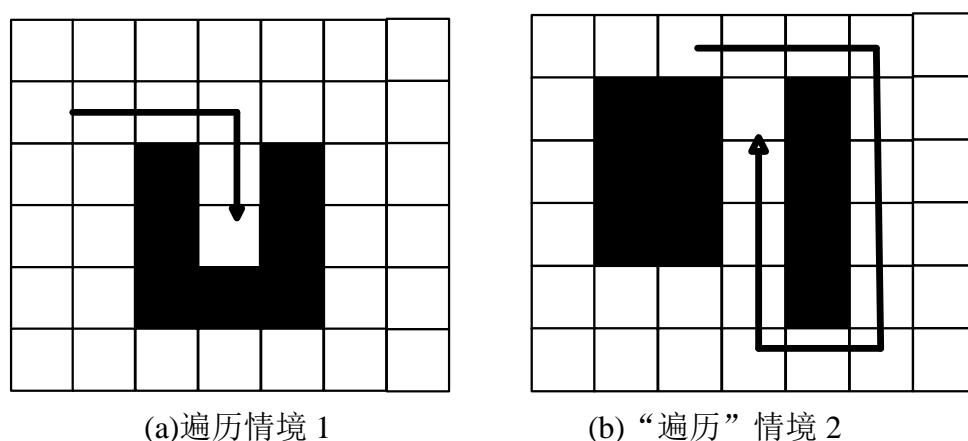


图 4.1 遍历陷入“死角”示意图

Fig 4.1 Schematic diagram of traversing into a "dead corner"

4.3 典型遍历算法

适用无人船的遍历算法包括随机式遍历算法、螺旋式遍历算法、往返前进式遍历算法等。

4.3.1 随机式遍历算法

随机式遍历算法^[54]最初作为一种启发式算法,一开始用于处理机器人简单的路径规划问题,使机器人按直线进行移动,当机器人遇到障碍物时,会主动避开障碍物。通过随机转动前进方向后继续行走直至绕开障碍物,并重新进行直线行走直到覆盖目标区域。

随机式遍历算法实现简单,无需复杂的路径规划算法,如果工作环境区域较小执行搜索任务采用随机式遍历式算法是一种快速解决方案。但随机式遍历算法局限性大,难以保证搜寻到最优路径、由于遇到障碍物时转向是随机的,所以会出现重复遍历相同的区域或在环境地图中某些区域没有搜索过的情况、搜索效率低。

4.3.2 螺旋式遍历算法

螺旋式遍历算法的思想是无人船从起始点出发，沿着一条螺旋线路径逐渐向内或者向外扩展做螺旋式运动直至遍历整个目标区域。根据无人船的初始和最终的运动位置与环境地图中心位置的欧几里得距离差值是否为正值，分为内螺旋式和外螺旋式遍历算法。在进行螺旋式遍历算法时，无人船沿固定方向进行移动，行动路径轨迹为螺旋线。当无人船前方探测到的区域为障碍物，则向左或向右转动 90° 继续前进；如果无人船探测到的前方区域为安全区域且未搜索过，则沿原方向继续前进。若初始位置与中心位置的欧几里得距离和遍历停止位置与中心位置的欧几里得距离的差值为正值，则为内螺旋遍历，若为负值则为外螺旋遍历。螺旋式遍历算法是由 Butler 等人提出的一种基于模板法的完全覆盖遍历算法^[55]。

螺旋式遍历算法规划的路径紧凑，覆盖全面，在大规模目标区域中效率较高，重复遍历相同区域的概率小，不需要多次往返，节省时间。但是对于具有复杂障碍物布局的目标区域可能不够适用，因为螺旋式遍历算法未考虑在复杂障碍物布局的目标区域避让策略，例如“死角”情境。

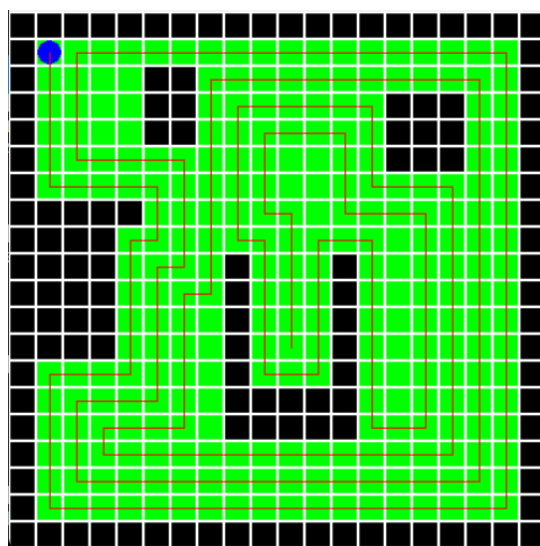


图 4.2 内螺旋式遍历算法示意图

Fig 4.2 Schematic diagram of inner spiral traversal algorithm

4.3.3 往返前进式遍历算法

往返前进式遍历算法是无人船在规划的工作区域内从起始点开始，沿着一条路径向前直线行走，直到到达工作区域的边缘。然后，无人船以相反的方向返回与起始船

位的横坐标相同的位置，每次转向按 90° 进行转向，在碰到边缘区域时先转向 90° 行驶一个船位时停止，然后再转向 90° 继续前进，如此反复往返遍历搜索，直至无法行驶或者遍历完目标区域^[56]，搜索停止。

往返前进式遍历算法不需要复杂的路径规划，实现相对简单，但是往返前进式遍历算法可能存在路径不够紧凑的情况，效率相对较低。对于大规模的目标区域，可能需要多次往返才能完成遍历，算法性能较低，且存在重复率大或者没有完全遍历目前区域的情况，算法搜索时间较长。

这三种遍历算法都有其独特的工作原理和研发起源，每种算法都试图在简单性、效率性和覆盖全面性之间找到一种平衡^[57]。其中螺旋式遍历算法相比于随机式遍历算法和往返前进式遍历算法更注重覆盖全面性和算法搜索效率，而随机式遍历算法和往返前进式遍历算法强调简单性和直观性。在本文中，重点研究无人船在栅格化后的海图中如何实现完全遍历和尽快搜索至目标点。所以选择螺旋式遍历算法更能体现出算法优势，不仅可以完全遍历，而且算法重复率低，减少了搜索路径长度，适合在较大区域范围搜索目标。

4.4 基于改进 A*的完全遍历路径规划

4.4.1 改进双向 A*算法

双向 A*算法是在传统的 A*算法的基础上进行改进的一种路径规划算法^[58]，传统的 A*算法从起始点开始搜索至终点，然而这种单向搜索的方式在搜索效率上存在着一定的局限性。相比之下，双向 A*算法采用从起点和终点同时开始搜索的策略，即从起点向终点进行搜索，同时从终点向起点进行搜索，并将搜索节点储存在两边的封闭链表中。直至两边的封闭链表中出现相同的搜索节点，意味着出现相同搜索节点，搜索过程结束。双向 A*算法相比于单向搜索的 A*算法搜索效率更高，极大的缩短了搜索路径的时间。特别是在搜索空间相对较大的情况下^[59]。

双向 A*算法的评价函数为

$$\begin{aligned} f_s(n) &= g_s(n) + h_s(n) \\ f_e(n) &= g_e(n) + h_e(n) \end{aligned} \quad (4.2)$$

式(4.2)中 $f_s(n)$ 代表从起始节点到节点 n 的估计总代价； $f_e(n)$ 代表从节点 n 到目标节点的估计总代价； $g_s(n)$ 代表从起始节点到节点 n 的实际代价。 $g_e(n)$ 代表从目标节点到节点 n 的实际代价。 $h_s(n)$ 代表从节点 n 到目标节点的启发式估计代价。 $h_e(n)$ 代表从节点 n 到起始节点的启发式估计代价。

双向 A*算法通过利用对比两个方向上的储存节点的信息,能够更快地收敛于一个共同的节点,从而减少算法的搜索的空间,提高了搜索效率,特别是在栅格地图尺寸较大或者复杂地形中。双向 A*算法由于从起点和终点同时搜索,因此可以更准确地找到最优路径,可以提高路径规划的准确性和可靠性。在本论文中,在双向 A*算法探索中,当两边探索的时候有一边或者两边都遇到“死角”问题,可以用改进后的 A*算法跳出死角,继续进行双向搜索直至终点。本文改进双向 A*算法步骤如下:

(1)首先对起始点和目标点各自建立两个不同的 *OpenList* 链表和 *CloseList* 链表,一个 *OpenList* 链表设置为从起始点向目标点搜索过程中存储未搜索过节点信息的正向链表,另一个 *OpenList* 链表设置为从目标点向起始点搜索未探寻过节点信息的反向链表。起始点 S 放入正向 *OpenList* 链表中,目标点 E 放入反向 *OpenList* 链表中,同时从两端开始搜索。两个 *CloseList* 链表均置为空链表,一个存储从起始点向目标点搜索过程中已经遍历过的节点,一个存储从目标点向起始点搜索过程中已经遍历过的节点。

(2)同时由目标点和起始点向各自的方向进行搜索,搜索方法和改进后的 A*算法一致。正向和反向 *OpenList* 链表在搜索过程中均选取估计代价成本 $f(r_i)$ 最小的节点作为各自当前节点。若该节点不在 *OpenList* 链表中,则将其加入各自的 *OpenList* 链表中,并把当前节点作为父节点。并在遍历过该节点后将其加入各自的 *CloseList* 链表中。若该节点在 *OpenList* 链表中,则比较该点的估计代价成本 $f(r_i)$ 与 *OpenList* 链表中的 $f(r_i)$ 值。若小于链表中的值,则用当前 $f(r_i)$ 值替换 *OpenList* 链表中的 $f(r_i)$ 值。若该节点在 *CloseList* 链表,则跳过该节点。

(3)当两边在进行路径搜索过程中遇到“死角”情况时,用改进后的 A*算法跳出“死角”,并继续步骤(2)。

(4)当正向 *OpenList* 链表和反向 *OpenList* 链表为空时,表示已完全遍历,退出循环,若双向搜索的某一个 *OpenList* 链表中的节点出现在其中一个 *CloseList* 链表中,则表示此时完成路径搜索,得到最终路径,算法结束。

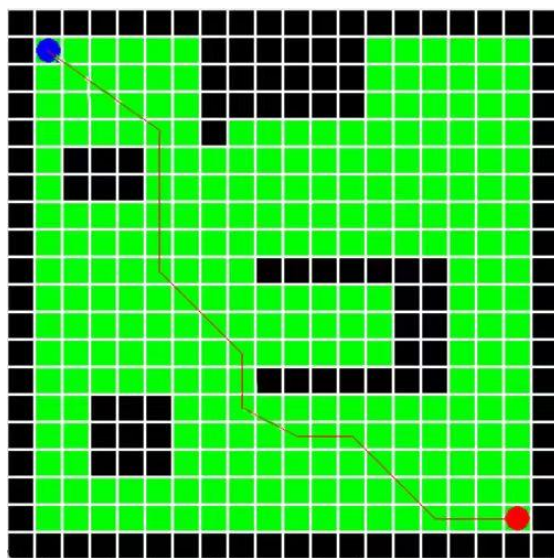


图 4.3 改进双向 A*算法示意图

Fig 4.3 Schematic diagram of improving the bidirectional A* algorithm

4.4.2 改进双向 A*算法的无人船完全遍历算法

改进双向 A*算法的无人船完全遍历路径规划算法是将改进双向 A*算法与内螺旋算法相结合。当无人船在海上执行搜救任务时, 确定好搜救区域并栅格化从船舶起始点至目标点的相关海图区域, 将无人船出发位置设为起始点, 失事船舶设为目标点。

当无人船在栅格化后的海图中执行完全遍历时，首先执行双向 A*算法找到从起始点至目标点的规划路径。当无人船在双向 A*算法搜索过程中陷入“死角”时，通过改进后的 A*算法跳出死角，继续遍历。

4.5 本章小结

本章首先探讨无人船路径规划的核心问题和“死角”情景,接着通过对衡量算法性能的几个指标分析对比了几种常见的路径搜索方法和完全遍历算法,根据本文的实际需求选择螺旋式遍历算法。然后详细阐述了传统 A*算法的基本概念和算法原理,并通过对 A*算法的不足指出提出了改进 A*算法。在栅格地图中进行路径规划结合算法实验数据实际说明了改进 A*算法的过程和优点。紧接着提出双向 A*算法,阐述了双向 A*算法相比于 A*算法更适合在无人船任务环境较大的区域进行路径规划,指出了双向 A*算法优势所在,通过改进 A*算法的原理来改进双向 A*算法,当双向 A*算法路径搜索过程中遇到“死角”时可以跳出死角,并提出一种改进双向 A*算法的无人船完全遍历路径规划算法,第五章将通过在栅格化的海图中进行对比实验来验证该算法的有效性。

5 算法实验验证和结果分析

本文第三章和第四章分别对无人船栅格法环境建模和无人船完全遍历路径规划算法进行了详细说明和改进,本章节将通过在栅格化后的电子海图里进行改进后的完全遍历路径规划算法实验来对比分析以验证算法的优点。最后对改进后的完全遍历路径规划算法进行路径跟踪实验来证明算法在无人船实际航行环境中的可行性。

5.1 海图栅格建模及算法实验过程

由本文第四章可知,双向 A*算法搜索方向是同时从起始点和目标点开始搜索,并在搜索过程中相互交汇,尤其是在大型地图或者起始点与目标点之间存在较长的路径时算法路径规划表现比 A*算法更好,减少了搜索空间和算法搜索时间。

为了进一步验证算法的有效性,将选择环境较为复杂的舟山群岛水域作为无人船仿真环境区域。建模方法与第三章海图栅格法环境建模相同,首先利用图像处理方法将 ECDIS 工具获取的原始舟山海域海图,如图 5.1(a)。并将其二值化处理,如图 5.1(b)。其中舟山群岛海域较为复杂,根据无人船实际吃水将陆地、浅水域、浅滩、航道线和碍航物处理成不可航行区域,其余的处理为可航区域。并对二值化后的海图做进一步优化使其更符合无人船工作环境,如图 5.2(a)。

并使用 MATLAB 将提取出的海图特征数据绘制成栅格地图如图 5.2(b)。

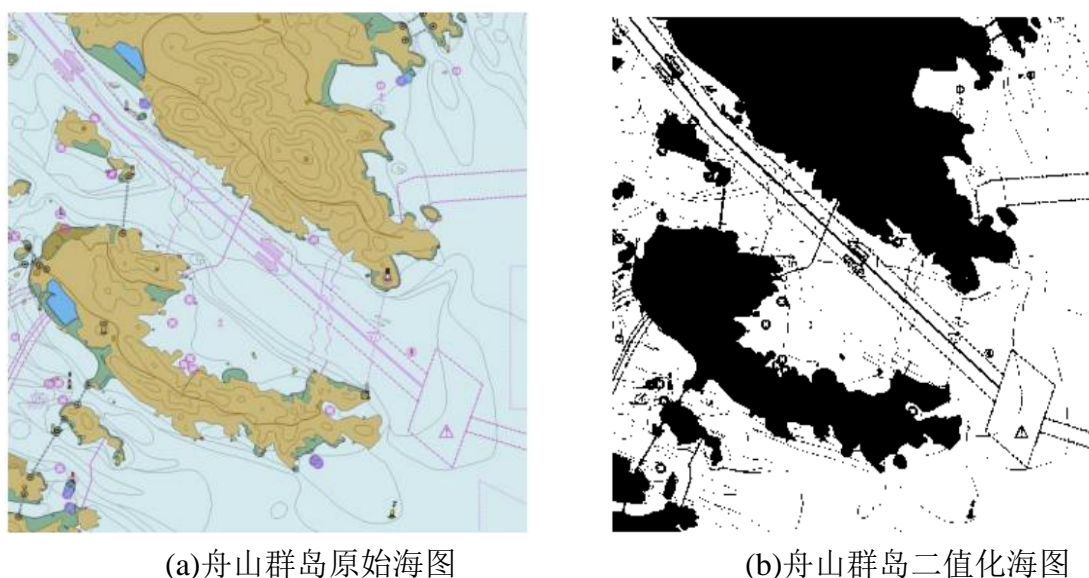


图 5.1 舟山群岛电子海图

Fig 5.1 Electronic nautical chart of Zhoushan Islands

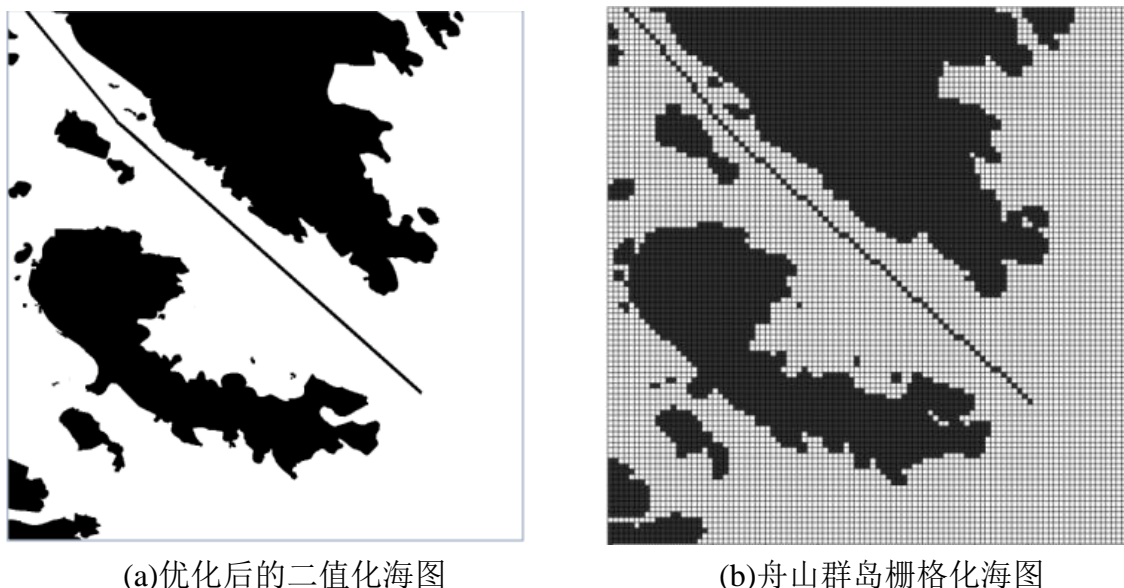


图 5.2 舟山群岛栅格地图

Fig 5.2 Zhoushan islands Raster Map

在本章节，首先将选取的海图二值化后进行栅格化处理，舟山群岛海图被处理成一个分辨率为 100×100 的栅格地图，舟山群岛海图的四个边缘角的坐标分别为经度（ $122^{\circ} 12.995\text{E} \sim 122^{\circ} 35.2633\text{E}$ ）、纬度（ $29^{\circ} 71.942\text{N} \sim 29^{\circ} 83.554\text{N}$ ）。

根据是否进行完全遍历选取不同的算法流程，当算法在搜索过程中陷入“死角”，可以通过改进 A*算法跳出。实验流程如图 5.3 所示。

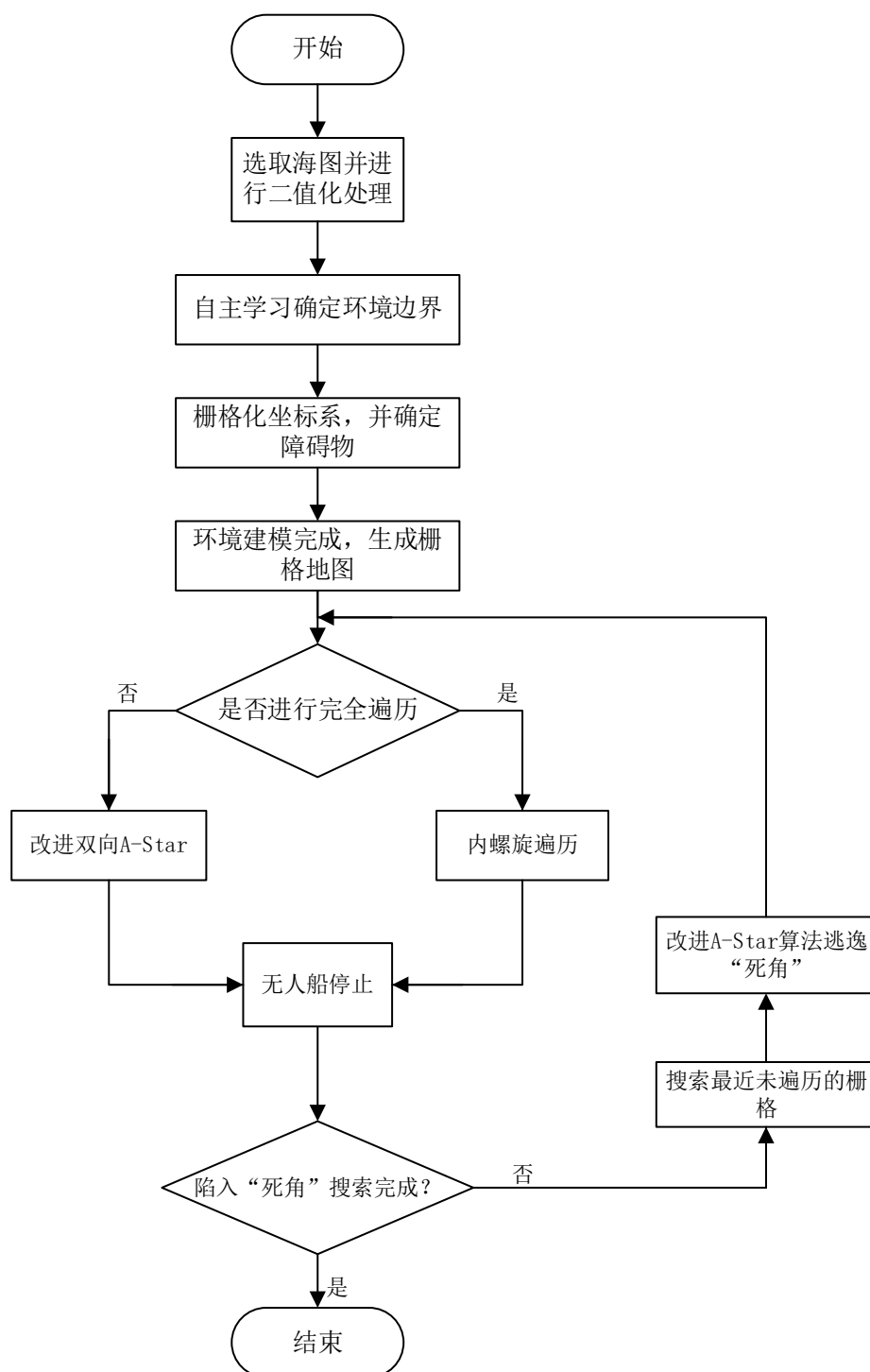


图 5.3 实验过程示意图

Fig 5.3 Schematic diagram of the experimental process

5.2 算法对比实验

第四章已经介绍过传统算法对比实验 A*算法、改进 A*算法和改进双向 A*算法的基本原理，为了更清晰的表述双向 A*算法相比于传统 A*算法和改进 A*算法在搜索空间较大的情况下，改进双向 A*算法能提高搜索效率，减少搜索路径步数上的优势，将利用第三章栅格法建模原理对舟山群岛海图进行模拟仿真并建立栅格环境模型。无人船的起始位置坐标为（29.8122° N，122.2210° E），栅格化后的海图尺寸为 152*152 的矩形。由于进行搜索算法之间的对比实验，所以不用进行完全遍历，设置目标点，目标点的位置坐标为（29.7281° N，122.2646° E）。海图中的深色区域为障碍物区域，绿色区域为可行区域，灰色区域为算法搜索过的区域，对传统 A*算法、改进双向 A*算法进行在栅格化后的海图里的路径规划实验。每一个栅格看成边长为1的正方形方格，将无人船的体积大小看成一个正方形方格，每个方格的所占的纬度和经度的大小为（0.000764N，0.001465E）

首先对传统 A*算法进行路径规划，如下图 5.4 所示。

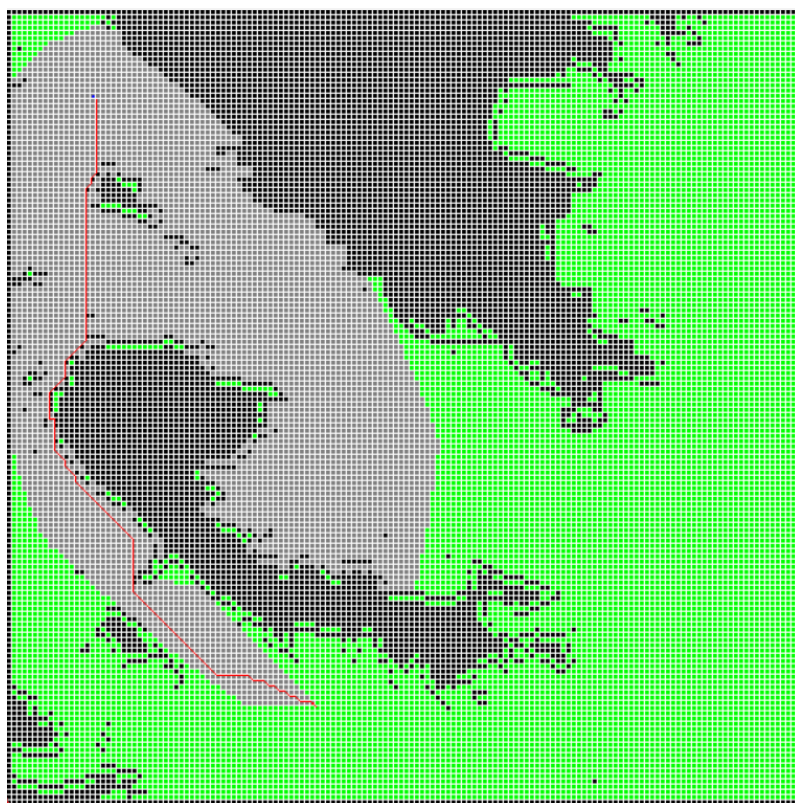


图 5.4 传统 A*算法路径规划示意图

Fig 5.4 Schematic diagram of traditional A* algorithm path planning

改进双向 A*算法在栅格化的舟山海图内路径规划实验，如下图 5.5 所示。

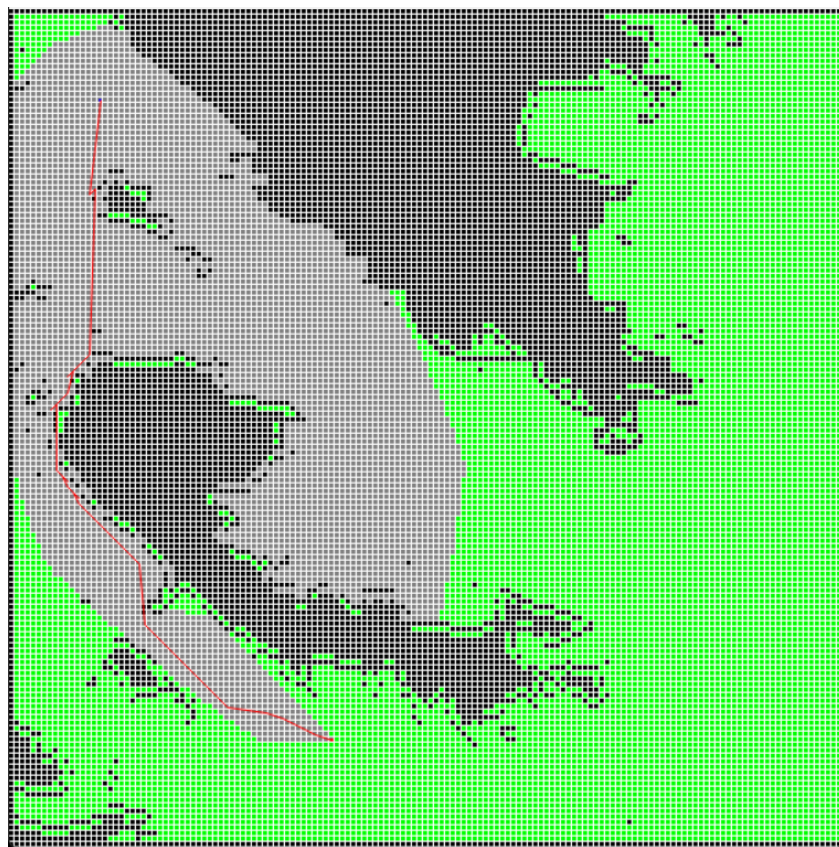


图 5.5 改进双向 A*算法路径规划示意图

Fig 5.5 Schematic diagram of path planning for improved bidirectional A* algorithm

表 5.1 传统 A*算法与改进双向 A*算法数据分析

Tab 5.1 Data Analysis of Traditional A* Algorithm and Improved Bidirectional A* Algorithm

两种算法	节点数	路径长度	算法运行时间/s
传统 A*算法	29	131	17.036
改进双向 A*算法	13	53	17.191

由表 5.1 可知，相比于传统 A*算法，改进双向 A*算法路程长度为 53，

路径节点数为 13，算法运行时间 17.191，而传统 A*算法路程长度为 131，路径节点数为 29，算法运行时间 17.036。改进后的双向 A*算法路径更加平滑，转折点更少，当遇到“死角”时可以跳出“死角”，减少了重复遍历率。

5.3 完全遍历路径规划实验

无人船在参与海上救援时往往需要先到达事故船舶所在位置进行救援，然后将事故船上人员送回起始点位置，再根据栅格化海图划分一定区域，在该区域内进行完全遍历，进行搜救幸存者或执行搜救任务。本节将从这个目的出发，进行完全遍历路径规划，以栅格化后的海图中心位置作为内螺旋遍历算法的圆心，该中心位置的坐标为： $(29.77.748^{\circ} \text{ N}, 122.24.129^{\circ} \text{ E})$ ，由于地图栅格化分辨率较高，代码运行数据较大，设置算法遍历停止点的坐标为 $(29.7837^{\circ} \text{ N}, 122.3147^{\circ} \text{ E})$ ，对内螺旋加改进双向 A* 算法进行完全遍历路径规划，当算法从起始点运行至停止点的规划路径如图 5.6。

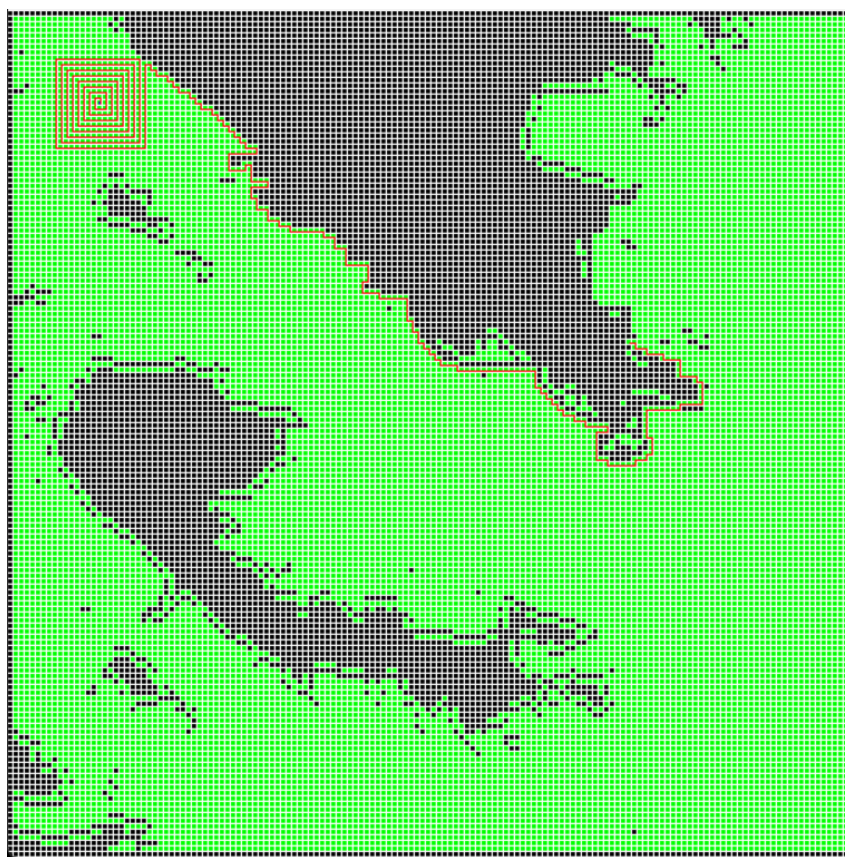


图 5.6 内螺旋加改进双向 A* 算法路径示意图

Fig 5.6 Schematic diagram of the path of the improved bidirectional A* algorithm with inner spiral

5.4 路径跟踪实验

路径跟踪是路径规划算法规划的路径使无人船可以完成各种任务的基础，也是验证无人船在海上自主航行轨迹是否符合无人船真实运动的重要方法^[60]，本文将基于第

二章无人船运动模型和 LOS 制导策略原理，与 PID 航向控制理论相结合在 MATLAB/Simulink 平台上搭建无人船路径跟踪仿真模型。对提出的内螺旋加改进双向 A*算法进行无人船路径跟踪模拟仿真，以验证本文所提算法的可行性，本实验将在没有外界干扰的环境中进行。选取图 5.6 中的二组区域，设为(a)区域和(b)区域，如下图 5.7 所示。

本实验采用 LOS+PID 策略对无人船进行路径跟踪仿真实验，初始条件设置如下：起始位置为 $(0, 0)$ ；起始航向为 30° ；航速为 0.75m/s ，在无干扰情况下分析无人船在栅格化电子海图内的路径跟踪效果，MATLAB/Simulink 平台上的栅格大小为一个栅格距离为 50m ，设无人船在栅格化的电子海图上行驶的一个单位栅格距离为 30m 。

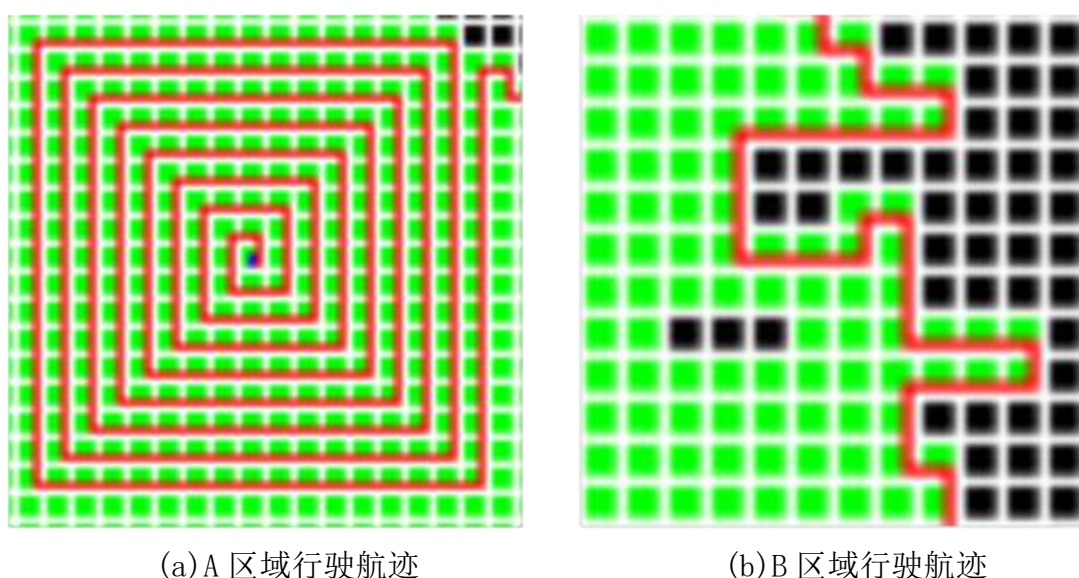


图 5.7 无人船行驶不同区域的航迹

Fig 5.7 The trajectory of unmanned ships traveling in different areas

首先对 A 区域无人船行驶航迹进行路径跟踪，建立二维直角坐标系，以正北和正东方向为正方向。航迹点的坐标信息如下表 5.2 所示。对航迹点的设定完成后，通过 MATLAB/Simulink 仿真得到无人船在 A 区域的航迹跟踪如下图 5.8 所示。

表 5.2 A 区域转向点及其坐标

Tab 5.2 The turning point and its coordinates in Area A

航迹转向点	X 轴	Y 轴
1	0	0

2	0	30
3	-30	30
4	-30	-30
5	30	-30
6	30	60
7	-60	60
8	-60	-60
9	60	-60
10	60	90
11	-90	90
12	-90	-90
13	90	-90
14	90	120
15	-120	120
16	-120	-120
17	120	150
18	-150	150
19	-150	-150
20	150	-150
21	150	180
22	-180	180
23	-180	-180
24	180	-180
-25	180	210
26	-210	210
27	-210	-210
28	210	-210
29	210	240
30	-240	240
31	-240	-240
32	240	-240
33	240	270

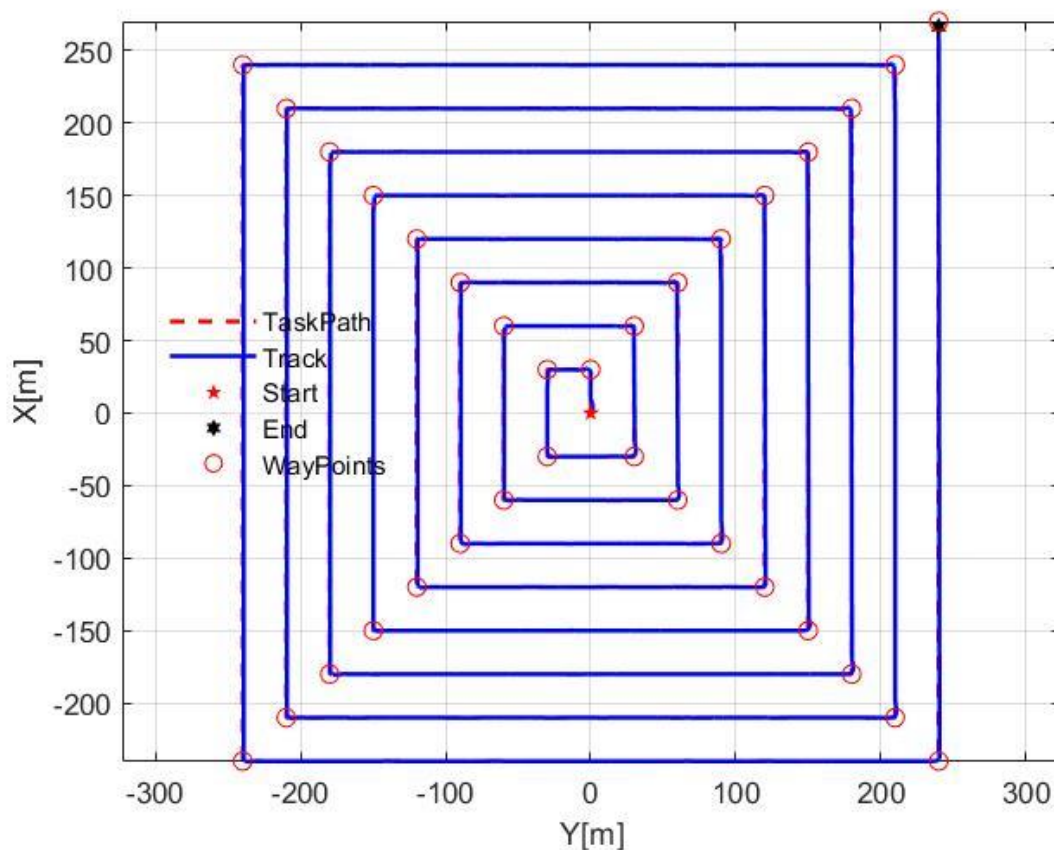


图 5.8 A 区域规划航迹点航迹跟踪控制验证

Fig 5.8 Verification of Track Tracking Control for Plan Track Points in Area A

对 B 区域无人船行驶航迹进行路径跟踪，建立二维直角坐标系，以正北和正东方向为正方向。航迹点的坐标信息如下表 5.3 所示。对航迹点的设定完成后，通过 MATLAB/Simulink 仿真得到无人船在 B 区域的航迹跟踪如下图 5.9 所示。

表 5.3 B 区域转向点及其坐标

Tab 5.3 The turning point and its coordinates in Area B

航迹转向点	X 轴	Y 轴
1	0	0
2	0	-30
3	30	-30
4	30	-60
5	90	-60
6	90	-90
7	-60	-90

8	-60	-180
9	30	-180
10	30	-150
11	60	-150
12	60	-240
13	150	-240
14	150	-270
15	60	-270
16	60	-330
17	90	-330
18	90	-360

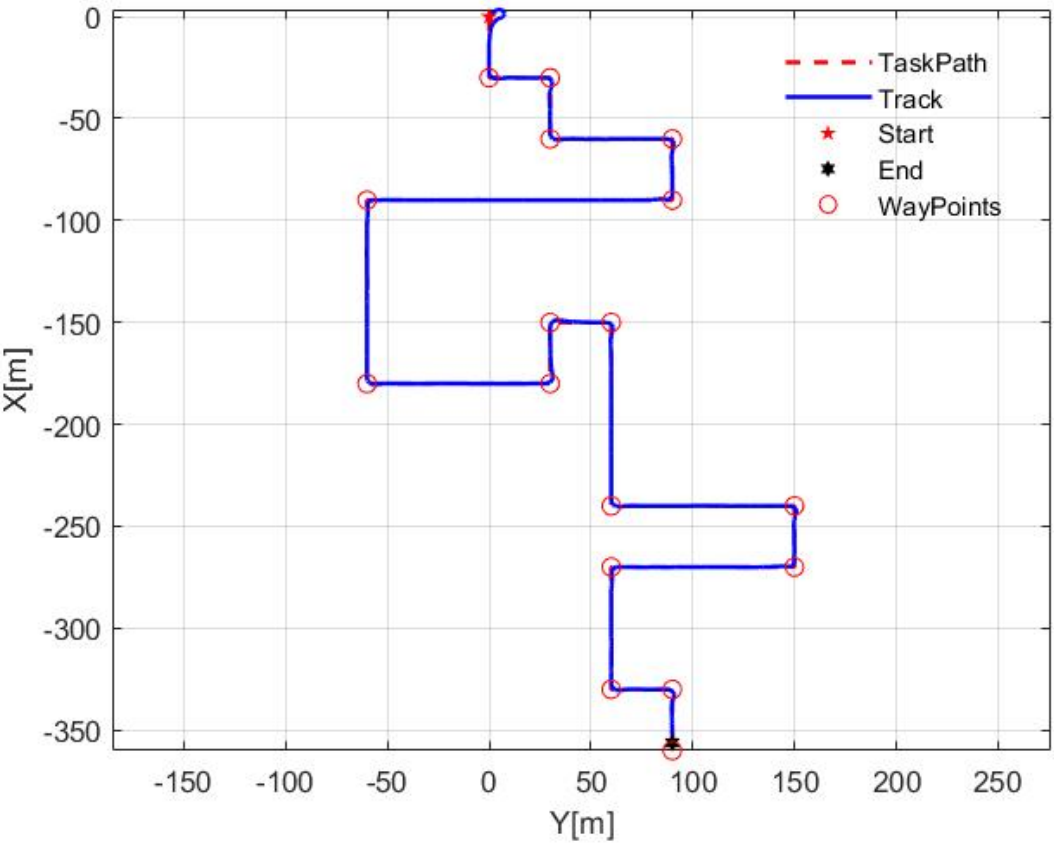


图 5.9 B 区域规划航迹点航迹跟踪控制验证

Fig 5.9 Verification of Track Tracking Control for Plan Track Points in Area B

通过对 A、B 区域内的无人船航迹跟踪控制仿真实验证明了算法的可行性，同时验证了 LOS 与 PID 算法策略对无人船航迹跟踪的准确性和稳定性。

5.5 本章小结

本章基于第三章栅格法环境建模的基础，对舟山海图相关区域进行栅格法环境建模，先把舟山海图处理成二值化并优化处理二值化后的海图，对二值化优化后的海图进行栅格化处理。并确定栅格化后的海图的四个边缘角的位置坐标。通过第四章对传统A*算法的进行实验，验证了传统A*算法规划出的路径不符合无人船实际运动，因此提出改进双向A*算法。并结合完全遍历算法，设计出了实验流程。确定了无人船的起始位置坐标和目标点的位置坐标及栅格化后的海图中心坐标和每一个栅格的尺寸。并通过在栅格化后的舟山海图内进行A*算法和改进双向A*算法的路径规划，不仅验证了改进双向A*算法的优点，还验证了内螺旋加改进双向A*算法的可行性和有效性

6 结论与展望

6.1 结论

本文主要研究了无人船完全遍历路径规划, 电子海图环境建模等方面的内容。通过对现有的全局路径规划算法和局部路径规划算法进行优点和不足的分析, 并对传统 A*算法进行改进, 使算法搜索更高效, 规划路径更平滑。对传统 DWA 算法进行改进, 通过引入模糊控制方法, 使 DWA 算法与模糊控制相结合, 通过分析实时环境信息与无人船的运动状态来调整评价因子的权重, 可以根据实际的情况进行动态调整, 提高路径规划的性能和鲁棒性, 以适应不同的场景和任务需求。

然后对完全遍历算法进行介绍并对比分析算法的优点和不足, 最终选择内螺旋算法。将内螺旋算法与改进双向 A*算法相结合, 根据是否进行完全遍历进行不同的算法流程。最后对电子海图进行栅格化处理, 在栅格化建模完成的海图地图上完成遍历路径规划。

在电子海图栅格法建模方面, 对电子海图进行合理的二值化阈值赋值, 减小栅格化后的电子海图与无人船真实航行环境的误差。对选定的舟山电子海图的边缘进行处理, 使其变成障碍物, 采集无人船工作的区域边缘的四个位置坐标及栅格化后的电子海图的中心位置坐标。把海图内的障碍物处理成障碍物栅格行驶, 将无人船的任务区域处理成栅格地图。确定无人船的起始点位置坐标和目标点的位置坐标, 通过对传统 A*算法和改进双向 A*算法在栅格任务区域内进行路径规划实验, 实验证明了改进双向 A*算法规划路径的节点数目比传统 A*算法少 16 个, 路径比传统 A*算法减少了 78 步。实验表明了改进双向 A*算法比传统 A*算法更高效。针对无人船在复杂环境下完全遍历路径规划算法效率低、普适性差, 遇到“死角”情境无人船遍历重复率高的问题, 提出了一种基于改进双向 A*算法的无人船内螺旋完全遍历算法。该算法首先通过判断是否进行完全遍历, 如果不进行完全遍历则通过改进双向 A*算法进行路径规划至目标点, 如果进行完全遍历, 则首先通过内螺旋完全遍历算法开始遍历, 当遍历遇到“死角”时, 利用水波算法搜索最近的未遍历的栅格, 通过改进双向 A*算法跳出“死角”, 继续进行路径搜索。通过实验表明, 算法可行有效。

6.2 展望

本文主要对完全遍历路径规划算法面临的一些问题进行研究。当无人船在海上执行救援任务时, 往往把失事船舶或者失事地点当作目标点, 先执行改进双向 A*算法, 到达目标区域进行救援。再规定相关航行区域用内螺旋算法做完全遍历, 具有一定的

价值与意义。但是文章中还存在着诸多不足之处以及可以改进的地方，在后续的研究工作中应当重视以下内容：

（1）本研究是在对无人船所航行区域的电子海图进行栅格化处理后的栅格地图上进行的，由于没有真实的无人船进行仿真实验，所以在实际航行中的数据和评价指标无法准确测量，算法实际结果无法达到理论上的要求。

（2）若使用真实的无人船进行研究，则无人船的传感器和导航定位装置环境数据采集和信息处理必须精确。要准确识别并测量障碍物的大小和无人船与障碍物之间的距离，并确定无人船及目标点的坐标位置信息。由于本研究是在静态环境下的栅格地图里进行船舶路径规划，并没有考虑无人船在动态环境里的路径规划。动态环境下影响无人船航行的因素更多，如何实现在复杂水域内无人船的动态路径规划是可以深入研究的方向。

（3）本文的无人船运动模型建立在船舶三自由度的基础上，并没有考虑无人船航行时的水下环境情况。可以在无人船上加装水下障碍物探测装置，探测水下环境信息，并将其反馈给无人船信息处理系统进行处理。本文栅格地图都是将障碍物在水平面以上的部分进行栅格化，在无人船实际航行时，无人船的水下区域环境更加复杂，障碍物的水下环境和周边环境可能无法使无人船安全航行，如何更清晰的表达无人船的可航行区域是未来值得研究的领域。

（4）实验中有些数据无法通过理论进行准确测量，无人船在电子海图栅格化后的地图上进行路径规划实验时，并没有考虑无人船在实际运动中转向的时间和无人船转向时所受的水体阻力和惯性力对无人船的能源消耗。

参考文献

- [1] Yan R, Pang S, Sun H, et al. Development and missions of unmanned surface vehicle[J]. Journal of Marine Science and Application, 2010, 9: 451-457.
- [2] Campbell S, Naeem W, Irwin G W. A review on improving the autonomy of unmanned surface vehicles through intelligent collision avoidance manoeuvres[J]. Annual Reviews in Control, 2012, 36(2): 267-283.
- [3] Liu Z, Zhang Y, Yu X, et al. Unmanned surface vehicles: An overview of developments and challenges[J]. Annual Reviews in Control, 2016, 41: 71-93.
- [4] 李家良.水面无人艇发展与应用[J].火力与指挥控制,2012,37(06):203-207.
- [5] Corfield S J, Young J M. Unmanned surface vehicles-game changing technology for naval operations[J]. IEE Control Engineering Series, 2006, 69: 311.
- [6] 文凤.美国海军研制高速无人船[J].国际航空,2002,(06):26.
- [7] Sherman J. Navy shifts Sea Hunter to advance USV development, fleet familiarization, CONOPS[J]. Inside the Navy, 2020, 33(2): 1-7.
- [8] Olejnik A. Trends in the development of unmanned marine technology[J]. Polish Hyperbaric Research, 2016, 55(2): 7-28.
- [9] 魏新勇.水面无人艇自主局部避障系统关键技术研究[D].华南理工大学,2019.
- [10] 张喜超. 基于 RRT*的无人船舶进出港自动路径规划研究[D]. 辽宁:大连海事大学,2022.
- [11] Svec P, Schwartz M, Thakur A, et al. Trajectory planning with look-ahead for unmanned sea surface vehicles to handle environmental disturbances[C]//2011 IEEE/RSJ International Conference on Intelligent Robots and Systems. IEEE, 2011: 1154-1159.
- [12] 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.
- [13] 陈超,唐坚. 基于可视图法的水面无人艇路径规划设计[J]. 中国造船, 2013 (1): 129-135.
- [14] 郭峰. 小型 USV 高速运动条件下避碰规划方法研究[D]. 黑龙江:哈尔滨工程大学,2019.
- [15] Wang N, Gao Y, Zheng Z, et al. A hybrid path-planning scheme for an unmanned surface vehicle[C]//2018 Eighth International Conference on Information Science and Technology (ICIST). IEEE, 2018: 231-236.
- [16] Singh Y, Sharma S, Sutton R, et al. Feasibility study of a constrained Dijkstra approach for optimal path planning of an unmanned surface vehicle in a dynamic maritime environment[C]//2018 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC). IEEE, 2018:117-122.
- [17] 王文明. 无人船自主靠泊避碰决策与运动控制研究[D]. 辽宁:大连海事大学,2022.
- [18] Rajput U, Kumari M. Mobile robot path planning with modified ant colony optimisation[J]. International Journal of Bio-Inspired Computation, 2017, 9(2): 106-113.
- [19] 曾凡明, 曹诗杰. 无人水面艇自主控制系统设计与试验研究[J]. 中国造船, 2016, 57(2): 164-174.
- [20] Grisetti G, Stachniss C, Burgard W. Improved techniques for grid mapping with rao-blackwellized particle filters[J]. IEEE transactions on Robotics, 2007, 23(1): 34-46.

- [21] 庄佳园, 张磊, 孙寒冰, 等. 应用改进随机树算法的无人艇局部路径规划[J]. 哈尔滨工业大学学报, 2015, 47(1):
- [22] 展邦顺. 欠驱动水下机器人路径规划和跟踪控制研究[D]. 青岛科技大学, 2023.
- [23] 刘志强, 卞涛, 叶曦, 等. 无人艇航迹跟踪控制系统设计与验证[J]. 科学技术与工程, 2022, 22(30): 13527-13536.
- [24] Liu Z, Yu L, Xiang Q, et al. Research on USV trajectory tracking method based on LOS algorithm[C]//2021 14th International Symposium on Computational Intelligence and Design (ISCID). IEEE, 2021: 408-411.
- [25] 刘志强. 基于模型预测控制的无人艇航迹跟踪策略研究[D]. 江汉大学, 2023.
- [26] 张敏捷. 海洋环境的自主建模与无人船路径规划方法研究[D]. 浙江: 浙江大学, 2022.
- [27] Wang Y, Goila A, Shetty R, et al. Obstacle avoidance strategy and implementation for unmanned ground vehicle using LIDAR[J]. SAE international journal of commercial vehicles, 2017, 10(2017-01-0118): 50-55.
- [28] 朱青. 基于蚁群算法的船舶航线设计方法的研究[D]. 黑龙江: 哈尔滨工程大学, 2011. DOI:10.7666/d.y2053429.
- [29] 杜道轶. 面向通道避障的 Mecanum 轮式机器人路径规划设计与实现[D]. 重庆: 重庆大学, 2016.
- [30] 陈奕虎. 面向动态环境的室内移动机器人路径规划研究[D]. 广西: 广西师范大学, 2023.
- [31] Dijkstra E W. A note on two problems in connexion with graphs[M]//Edsger Wybe Dijkstra: His Life, Work, and Legacy. 2022: 287-290.
- [32] 谢法吾, 李玲玲, 李丽, 等. 基于多目标混合进化算法的作业车间混排可变分批节能调度方法[J]. 中国机械工程, 2023, 34(13): 1576-1588, 1598. DOI:10.3969/j.issn.1004-132X.2023.13.007.
- [33] Hart P E, Nilsson N J, Raphael B. Correction to "a formal basis for the heuristic determination of minimum cost paths"[J]. ACM SIGART Bulletin, 1972 (37): 28-29.
- [34] 周伟, 胡毅, 刘进江, 等. 基于 Dijkstra 算法的 AMR 集群协同调度研究[J]. 制造技术与机床, 2023, (06): 175-179.
- [35] 鲁毅, 高永平, 龙江腾. A*算法在移动机器人路径规划中的研究[J]. 湖北师范大学学报(自然科学版), 2022, 42(02): 59-65.
- [36] 柴荣. 基于贪心算法的物流配送系统设计与实现[D]. 甘肃: 西北师范大学, 2014. DOI:10.7666/d.D568198.
- [37] Chengjun Z, Xiuyun M. Spare A* search approach for UAV route planning[C]//2017 IEEE international conference on unmanned systems (ICUS). IEEE, 2017: 413-417.
- [38] Dorigo M, Maniezzo V, Colorni A. Ant system: optimization by a colony of cooperating agents[J]. IEEE transactions on systems, man, and cybernetics, part b (cybernetics), 1996, 26(1): 29-41.
- [39] 王沛栋. 改进蚁群算法及在路径规划问题的应用研究[D]. 中国海洋大学, 2012.
- [40] Liu Y, Ma J, Zang S, et al. Dynamic path planning of mobile robot based on improved ant colony optimization algorithm[C]//Proceedings of the 2019 8th international conference on networks, communication and computing. 2019: 248-252.
- [41] 熊超文. 蚁群算法的改进及其在路径规划中的应用研究[D]. 重庆: 重庆邮电大学, 2020.

- [42] 张松灿,普杰信,司彦娜,等.蚁群算法在移动机器人路径规划中的应用综述[J].计算机工程与应用,2020,56(08):10-19
- [43] 钟富川.基于 RRT 算法的无人艇运动规划方法研究[D].重庆交通大学,2023.
- [44] Taheri E, Ferdowsi M H, Danesh M. Fuzzy greedy RRT path planning algorithm in a complex configuration space[J]. International Journal of Control, Automation and Systems, 2018, 16: 3026-3035.
- [45] 林泽南. 基于全局改进势场与局部动态避障的移动机器人路径规划方法研究[D]. 辽宁:大连理工大学,2021.
- [46] 李长庚. 改进 A*算法和动态窗口法的移动机器人路径规划研究[D]. 重庆理工大学,2022.
- [47] Fox D, Burgard W, Thrun S. The dynamic window approach to collision avoidance[J]. IEEE Robotics & Automation Magazine, 1997, 4(1): 23-33.
- [48] Lima D A, Victorino A C. An image based dynamic window approach for local navigation of an autonomous vehicle in urban environments[C]//IEEE ICRA Workshop on Modelling, Estimation, Perception and Control of All Terrain Mobile Robots (WMEPC 2014). 2014.
- [49] 陈奕虎. 面向动态环境的室内移动机器人路径规划研究[D]. 广西:广西师范大学,2023.
- [50] 郑文博. 智能车辆局部路径规划与路径跟踪控制研究[D]. 陕西:西安科技大学,2022.
- [51] 包永峰. 基于改进蚁群融合算法的变电站巡检机器人路径规划研究[D]. 天津:天津理工大学,2023.
- [52] 张月.清洁机器人全覆盖路径规划研究[D].重庆大学,2015.
- [53] Wang M, Liu J N K. Fuzzy logic-based real-time robot navigation in unknown environment with dead ends[J]. Robotics and autonomous systems, 2008, 56(7): 625-643.
- [54] Zhang P, Xiong C, Li W, et al. Path planning for mobile robot based on modified rapidly exploring random tree method and neural network[J]. International Journal of Advanced Robotic Systems, 2018, 15(3): 1729881418784221.
- [55] 王琦斐,杨军.基于内螺旋覆盖算法的多 AUV 协作反水雷路径规划研究[J].计算机测量与控制,2012,20(01):144-146+160.
- [56] 高鹏宇.油菜无人播种测控管云平台设计与试验[D].华中农业大学,2022.
- [57] 王朝飞,王慎执,宋士吉,等.面向海域巡逻的多无人船路径规划方法及仿真[J].中国图象图形学报,2023,28(08):2536-2548
- [58] 方文凯,廖志高.基于改进 A*算法的移动机器人全局路径优化研究[J].广西科技大学学报,2023,34(01):73-78+84.
- [59] 谭成志. 融合改进人工势场的 A*算法在机器人全局路径规划中的应用研究[D]. 广西:广西大学,2022.
- [60] 申继鹏.基于航向制导与控制的无人艇航迹跟踪算法研究及实现[D].华东师范大学,2022.