



(12) 发明专利申请

(10) 申请公布号 CN 113311843 A

(43) 申请公布日 2021.08.27

(21) 申请号 202110649846.8

(22) 申请日 2021.06.10

(71) 申请人 大连海事大学

地址 116026 辽宁省大连市高新园区凌海路1号

(72) 发明人 张秀国 张凤阁 张英俊 王少博 曹志英 杜意权 魏岩旭

(74) 专利代理机构 大连东方专利代理有限责任公司 21212

代理人 李馨

(51) Int. Cl.

G05D 1/02 (2020.01)

权利要求书1页 说明书7页 附图6页

(54) 发明名称

基于安全距离约束和LOS视线判断的无人船路径规划方法

(57) 摘要

本发明提供一种基于安全距离约束和LOS视线判断的无人船路径规划方法,包括:获取船舶航行环境信息,并将所述环境信息转化为实验环境数据;获取起点坐标和终点坐标,并基于LOS视线算法判断起点和终点间的连线是否穿过障碍物;选取移动消耗F值最小的节点n作为当前节点,并基于LOS视线算法判断当前节点n和终点间的连线是否穿过障碍物;获取当前节点n的相邻节点,对每个相邻节点使用安全距离探测算法以判断其在安全距离d范围内是否存在障碍物;重复执行上述步骤,直到某一节点n与终点间的连线不穿过障碍物时算法结束,将规划路径中得到的路径节点按先后顺序进行连线,形成最终规划路线。



CN 113311843 A

1. 一种基于安全距离约束和LOS视线判断的无人船路径规划方法,其特征在于,包括:

S1、获取船舶航行环境信息,并将所述环境信息转化为实验环境数据,所述环境信息至少包括障碍物坐标信息;

S2、获取实验环境数据中的起点坐标和终点坐标,并基于LOS视线算法判断起点和终点间的连线是否穿过障碍物,若不穿过障碍物,则将起点和终点添加到路径末尾,执行S6,否则将起点加入到OPEN集合,执行S3,其中所述OPEN集合用于存储算法待遍历的节点;

S3、从所述OPEN集合中选取移动消耗F值最小的节点n作为当前节点,并基于LOS视线算法判断当前节点n和终点间的连线是否穿过障碍物,若不穿过障碍物,则将当前节点n与终点添加到路径末尾,执行S6,否则执行S4;

S4、获取当前节点n的相邻节点,对每个相邻节点使用安全距离探测算法以判断其在安全距离d范围内是否存在障碍物,若不存在障碍物则将该相邻节点加入到OPEN集合中,执行S3,否则使用惩罚函数对该相邻节点的移动消耗F值赋予惩罚值,使该节点在后续的遍历中不会被选取;

S5、重复执行上述S3到S4,直到某一节点n与终点间的连线不穿过障碍物时算法结束,路径规划完毕,执行S6;

S6、将规划路径中得到的路径节点按先后顺序进行连线,形成最终规划路线。

2. 根据权利要求1所述的基于安全距离约束和LOS视线判断的无人船路径规划方法,其特征在于,获取船舶航行环境信息,并将所述环境信息转化为实验环境数据,包括:

从电子海图中获取障碍物位置坐标信息,使用墨卡托转换方法将障碍物点坐标转换成直角坐标系下的坐标;

将转换后的坐标进行放缩,使障碍物坐标适应实验环境从而获得实验环境数据。

3. 根据权利要求1所述的基于安全距离约束和LOS视线判断的无人船路径规划方法,其特征在于,获取船舶航行环境信息,并将所述环境信息转化为实验环境数据,还包括:

基于实验环境数据生成栅格环境数据,所述栅格环境数据包括白色或者黑色的方格,其中,白色方格代表船舶可航行的区域,黑色方格代表船舶不可航行的障碍区域。

4. 根据权利要求1所述的基于安全距离约束和LOS视线判断的无人船路径规划方法,其特征在于,所述移动消耗F值包括从起点到当前节点的移动代价和当前节点到目标节点的估算成本。

基于安全距离约束和LOS视线判断的无人船路径规划方法

技术领域

[0001] 本发明涉及路径规划领域,具体而言,尤其涉及一种基于安全距离约束和LOS视线判断的无人船路径规划方法。

背景技术

[0002] 无人船作为开发和利用海洋、河道的重要工具之一,凭借其无人化、小型化和智能化的优点,正受到业界越来越多的关注。路径规划是保证无人船安全航行的基础和前提,在无人船研究领域占据着关键作用。

[0003] 无人船路径规划是在已知船舶航行环境下,快速规划一条安全有效的路径帮助船舶避开危险区域到达目标点。路径规划分为全局路径规划和局部路径规划。全局路径规划主要是针对已知环境中的已探明障碍物为船舶寻找到航行距离和时间最短的路径。局部路径规划是对无人船航行过程中突然出现的动态/静态障碍物的感知和避障,并且在完成局部路径规划之后能够自主回归到下一个航路点。

[0004] 目前,无人船路径规划方法主要包括传统的路径规划方法和仿生智能方法两类。传统的路径规划方法有人工势场算法(Artificial Potential Field,APF)、快速扩展随机树算法(Rapid-exploration Random Tree,RRT)、概率路图算法(Probabilistic Roadmap,PRM)等。APF算法通过将航行限制区转换为虚拟障碍物区域,构建虚拟势场避开静态障碍物,从而引导无人船驶向目标航路点,但是当有一个相等的斥力和引力时或当目标点的斥力很大时,船舶就会停滞不前,陷入局部最优。RRT算法通过状态空间的随机采样点,把搜索导向空白区域,从而寻找到一条从起始点到目标点的路径。该方法能够有效地解决高维空间和复杂约束的路径规划问题,但是该方法规划出来的路径通常不是最优的。PRM算法是一种基于图搜索的方法,该方法将连续空间转换成离散空间,再利用A*等搜索算法在路线图上寻找路径,以提高搜索效率。该方法能用相对其他随机类算法较少的采样点来找到一个解,对多数问题而言,相对少的样本足以覆盖大部分可行的空间并且找到路径的概率为1,但是当采样点过少或者分布不合理时,PRM算法是很难规划出路径的。

[0005] 仿生智能方法主要包括遗传算法(Genetic Algorithm,GA)、粒子群算法(Particle Swarm Optimization,PSO)和蚁群算法(Ant Colony Optimization,ACO)等。这类方法将船舶路径规划问题转换为路径优化问题。GA算法将适应度函数分成两部分,分别用来判断路径的长短和平滑程度,使用轮盘赌方法作为选择方法,通过选择、交叉和变异算子来寻找最优路径,但是该方法依赖于栅格环境,当栅格较大时,计算时间较长并且容易陷入局部最优。PSO算法通过更新个体学习因子、社会学习因子以及粒子的位置和速度来寻找最优路径,但是该方法存在易陷入局部最优、全局收敛性差等缺点。ACO算法以碰撞危险和航程损失作为目标函数来进行最优路径求解,但是该方法的计算量非常大并且对系统的性能要求非常高。

[0006] A*算法作为在静态路网中求解最短路径最有效的方法之一,自1968年诞生以来就得到了广泛的应用。原始A*算法先把要搜寻的区域划分成正方形的格子,从起点开始检查

其相邻的格子,逐渐向四周扩展直至找到目标,最后从可行的格子中找到移动代价最小的路径。但是该方法存在以下缺点:

[0007] (1) 原始A*算法规划出来的路径是贴着障碍物的,因此与船舶航行实际情况不符且存在极大航行风险。

[0008] (2) 原始A*算法必须遍历到目标节点算法才能结束,这在某些情况下可能存在冗余遍历。当海图环境较大时,需要耗费大量的空间来存储遍历的节点,同时也增加了算法的执行时间。

发明内容

[0009] 鉴于上述A*算法在无人船路径规划领域应用存在的问题,本发明提供基于安全距离约束和LOS视线判断的无人船路径规划方法。应用该方法规划出的路径始终与障碍物保持一定安全距离,符合船舶航行实际需求;且算法能够更快收敛,同时也降低了算法对存储空间的消耗。

[0010] 本发明采用的技术手段如下:

[0011] 一种基于安全距离约束和LOS视线判断的无人船路径规划方法,其特征在于,包括:

[0012] S1、获取船舶航行环境信息,并将所述环境信息转化为实验环境数据,所述环境信息至少包括障碍物坐标信息;

[0013] S2、获取实验环境数据中的起点坐标和终点坐标,并基于LOS视线算法判断起点和终点间的连线是否穿过障碍物,若不穿过障碍物,则将起点和终点添加到路径末尾,执行S6,否则将起点加入到OPEN集合,执行S3,其中所述OPEN集合用于存储算法待遍历的节点;

[0014] S3、从所述OPEN集合中选取移动消耗F值最小的节点n作为当前节点,并基于LOS视线算法判断当前节点n和终点间的连线是否穿过障碍物,若不穿过障碍物,则将当前节点n与终点添加到路径末尾,执行S6,否则执行S4;

[0015] S4、获取当前节点n的相邻节点,对每个相邻节点使用安全距离探测算法以判断其在安全距离d范围内是否存在障碍物,若不存在障碍物则将该相邻节点加入到OPEN集合中,执行S3,否则使用惩罚函数对该相邻节点的移动消耗F值赋予惩罚值,使该节点在后续的遍历中不会被选取;

[0016] S5、重复执行上述S3到S4,直到某一节点n与终点间的连线不穿过障碍物时算法结束,路径规划完毕,执行S6;

[0017] S6、将规划路径中得到的路径节点按先后顺序进行连线,形成最终规划路线。

[0018] 进一步地,获取船舶航行环境信息,并将所述环境信息转化为实验环境数据,包括:

[0019] 从电子海图中获取障碍物位置坐标信息,使用墨卡托转换方法将障碍物点坐标转换成直角坐标系下的坐标;

[0020] 将转换后的坐标进行放缩,使障碍物坐标适应实验环境从而获得实验环境数据。

[0021] 进一步地,获取船舶航行环境信息,并将所述环境信息转化为实验环境数据,还包括:

[0022] 基于实验环境数据生成栅格环境数据,所述栅格环境数据包括白色或者黑色的方

格,其中,白色方格代表船舶可航行的区域,黑色方格代表船舶不可航行的障碍区域。

[0023] 进一步地,所述移动消耗F值包括从起点到当前节点的移动代价和当前节点到目标节点的估算成本。

[0024] 较现有技术相比,本发明具有以下优点:

[0025] 本发明针对原始A*算法在进行无人船路径规划时存在的问题,对原始A*算法进行了两方面改进。一方面,在原始A*算法中加入了安全距离约束,使其规划出来的路径始终与障碍物保持一定的安全距离;另一方面,改进了原始A*算法的结束条件,即在原始A*算法中引入了LOS视线算法,去除了冗余遍历,使其在时间消耗和空间消耗方面都得到了优化。通过实验证明,本发明改进的A*算法不仅优于原始A*算法,而且与其他一些常用的无人船路径规划算法相比也具有明显优势。

[0026] 基于上述理由本发明可在无人船路径规划领域广泛推广。

附图说明

[0027] 为了更清楚地说明本发明实施例或现有技术中的技术方案,下面将对实施例或现有技术描述中所需要使用的附图做以简单地介绍,显而易见地,下面描述中的附图是本发明的一些实施例,对于本领域普通技术人员来讲,在不付出创造性劳动的前提下,还可以根据这些附图获得其他的附图。

[0028] 图1为本发明无人船路径规划方法流程图。

[0029] 图2为本发明栅格环境示意图。

[0030] 图3为原始A*算法路径规划结果示意图。

[0031] 图4为加入安全距离约束的A*算法(安全距离 $d=1$)路径规划结果示意图。

[0032] 图5为加入安全距离约束的A*算法(安全距离 $d=3$)路径规划结果示意图。

[0033] 图6为加入LOS视线算法与安全距离约束的A*算法路径规划结果示意图。

[0034] 图7为本发明算法规划的路径示意图。

[0035] 图8为A*算法规划的路径示意图。

[0036] 图9为APF算法规划的路径示意图。

[0037] 图10为RRT算法规划的路径示意图。

具体实施方式

[0038] 为了使本技术领域的人员更好地理解本发明方案,下面将结合本发明实施例中的附图,对本发明实施例中的技术方案进行清楚、完整地描述,显然,所描述的实施例仅仅是本发明一部分的实施例,而不是全部的实施例。基于本发明中的实施例,本领域普通技术人员在没有做出创造性劳动前提下所获得的所有其他实施例,都应当属于本发明保护的范围。

[0039] 需要说明的是,本发明的说明书和权利要求书及上述附图中的术语“第一”、“第二”等是用于区别类似的对象,而不必用于描述特定的顺序或先后次序。应该理解这样使用的数据在适当情况下可以互换,以便这里描述的本发明的实施例能够以除了在这里图示或描述的那些以外的顺序实施。此外,术语“包括”和“具有”以及他们的任何变形,意图在于覆盖不排他的包含,例如,包含了一系列步骤或单元的过程、方法、系统、产品或设备不必限于

清楚地列出的那些步骤或单元,而是可包括没有清楚地列出的或对于这些过程、方法、产品或设备固有的其它步骤或单元。

[0040] 如图1所示,本发明提供了一种基于安全距离约束和LOS视线判断的无人船路径规划方法,包括:

[0041] S1、获取船舶航行环境信息,并将所述环境信息转化为实验环境数据,所述环境信息至少包括障碍物坐标信息。

[0042] 具体来说,首先对障碍物进行预处理:

[0043] 首先从电子海图中获取障碍物位置坐标信息,使用墨卡托转换方法将障碍物点坐标转换成直角坐标系下的坐标,坐标转换公式如公式(1)所示;然后将转换后的坐标进行放缩,将障碍物坐标放缩到实验环境中。

$$[0044] \quad \begin{cases} x = (a * \theta) / 1000 \\ y = (a * \ln(\tan(\frac{\pi}{4} + \frac{\alpha}{2}) + \frac{e}{2} \ln(\frac{1 - e * \sin \alpha}{1 + e * \sin \alpha}))) / 1000 \end{cases} \quad (1)$$

[0045] 其中 $e = \frac{\sqrt{a^2 - b^2}}{a}$ 为地球椭球体的第一偏心率, a 为地球长轴, b 为地球短轴;地球上某点的经度为 θ , $\theta \in (-\pi, +\pi)$; 纬度为 α , $\alpha \in (-\pi/2, +\pi/2)$; (x, y) 为转换后的直角坐标。

[0046] 然后,将海图环境转化为A*算法可运行的由0、1元素组成的栅格环境,其中1元素(黑色方格)代表船舶不可航行的障碍,0元素(白色方格)代表船舶可航行的区域。栅格环境示意图如图2所示。

[0047] S2、获取实验环境数据中的起点坐标和终点坐标,并基于LOS视线算法判断起点和终点间的连线是否穿过障碍物,若不穿过障碍物,则将起点和终点添加到路径末尾,执行S6。否则将起点加入到OPEN集合,执行S3。其中所述OPEN集合用于存储算法待遍历的节点。

[0048] 具体来说,获得起点坐标与终点坐标,使用LOS视线算法判断起点和终点间的连线是否穿过障碍物,若两者连线不穿过障碍物则说明起点与终点是直线可达的,将终点添加到路径末尾,转到步骤6;若两者的连线穿过障碍物则将起点加入到OPEN集合(该集合用于存储A*算法中待遍历的节点),执行步骤3。其中使用LOS视线算法判断起点和终点间的连线是否穿过障碍物,包括:将栅格地图map,当前节点s,目标节点goal作为算法的输入,若节点s与目标节点goal之间有视线(即两点间连线不穿过障碍)返回1,反之返回0。

[0049] S3、从所述OPEN集合中选取移动消耗F值最小的节点n作为当前节点,并基于LOS视线算法判断当前节点n和终点间的连线是否穿过障碍物,若不穿过障碍物,则将当前节点n与终点添加到路径末尾,执行S6,否则执行S4。

[0050] 具体来说,从OPEN集合中选取F值最小的节点n移动到CLOSE集合中,并将该节点n作为要处理的当前节点。其中,F代表该当前节点的移动消耗(包含从起点到当前节点的移动代价和当前节点到目标节点的估算成本),CLOSE集合用于存储A*算法遍历过程中已经计算过的节点,该集合中存储的节点在算法的后续遍历中不会再被考虑。

[0051] 使用LOS视线算法判断当前节点n与终点间的连线是否穿过障碍物,若两者间的连线不穿过障碍物则说明当前节点n与终点是直线可达的,将当前节点n与终点添加到路径末尾,转到执行S6;若穿过障碍则执行S4。

[0052] S4、获取当前节点n的相邻节点,对每个相邻节点使用安全距离探测算法以判断其在安全距离d范围内是否存在障碍物,若不存在障碍物则将该相邻节点加入到OPEN集合中,执行S3,否则使用惩罚函数对该相邻节点的移动消耗F值赋予惩罚值,使该节点在后续的遍历中不会被选取。

[0053] 具体来说,获取当前节点n的相邻节点,对每个相邻节点使用安全距离探测算法来判断在安全距离d范围内是否存在障碍物,若不存在障碍物则将其相邻节点加入到OPEN集合中,执行S3;若存在障碍物则使用惩罚函数对该节点的F值赋予一个较大的惩罚值,使该节点在算法后续的遍历中,由于F值较大而不会被选取,从而实现改进的A*算法规划的路径始终与障碍物保持一定安全距离的目的。其中,基于安全距离探测算法来判断在安全距离d范围内是否存在障碍物,包括将栅格地图map,节点n的相邻节点s,安全距离d,惩罚值reward作为算法的输入,输出符合安全距离d要求的节点s。

[0054] S5、重复执行上述S3到S4,直到某一节点n与终点间的连线不穿过障碍物时算法结束,路径规划完毕,执行S6;

[0055] S6、将规划路径中得到的路径节点按先后顺序进行连线,形成最终规划路线。

[0056] 下面通过具体的应用实例,结合对比算法对本发明的方案和效果做进一步说明。

[0057] 本实施例中,

[0058] 在船舶航行过程中,整条路径被划分为若干个航路点,路径规划需要规划出一个航路点到下一个航路点之间的路径。如图3所示,在仿真环境中,设置环境大小为70像素×50像素,坐标(5,5)为路径起始点,坐标(65,45)为路径目标点,每两个航路点之间设置5个较大障碍物,并随机生成若干个较小障碍物,其中黑色部分代表障碍物,空白部分为可航行区域。

[0059] 分别从本发明改进后的A*算法与原始A*算法,以及本发明改进后的A*算法与其他路径规划常用算法,两个方面来验证本算法的有效性。

[0060] 改进后的A*算法与原始A*算法对比

[0061] A*算法是目前最常用的一种启发式路径探索算法,在众多的路径规划算法中有着显著的优势。然而随着海图环境变大时,A*算法需要遍历的节点数目是呈指数级增长的,这极大地增大了算法的计算量,而且需要耗费大量的空间来存储A*算法CLOSE集合中的节点。因此下面从算法规划的路径是否符合安全要求、拐点个数、运行时间和路径长度几个方面将本发明改进后的A*算法与原始A*算法进行对比来说明本发明的有效性。

[0062] 原始A*算法规划的路径如图3所示;在A*算法中加入安全距离约束并且设置安全距离d=1后规划的路径如图4所示;将安全距离设置为d=3时规划的路径如图5所示;同时加入安全距离约束和LOS视线算法后规划的路径如图6所示。算法对比情况如表3所示。

[0063] 表3本发明改进后的A*算法与原始A*算法对比

	是否符合安全要求	拐点个数/个	CLOSE集合节点个数/个	所用时间/s	路径长度/px
[0064] 原始A*算法	否	37	1690	0.078	79.49
本算法	是	1	234	0.031	78.79

[0065] 由表3可知,由本发明算法规划的路径是符合安全要求的,而A*算法规划的路径由于紧贴着障碍物是不符合实际需求的存在极大危险的路径。在拐点方面本发明算法的拐点

个数为1个,A*算法的拐点个数为37个。在存储空间方面本发明算法占用的CLOSE集合中节点个数为234个,A*算法为1690个。在运行时间方面本发明算法的运行时间为0.031s,A*算法的运行时间为0.078s。在路径长度方面本发明算法规划的路径长度为78.79像素,A*算法规划的路径为79.49像素。

[0066] 综上可知,在本发明算法规划的路径长度略优于A*算法的前提下,本发明算法在路径安全方面是优于A*算法的;在路径平滑方面本发明算法规划的路径拐点个数是1个,而A*算法规划的路径拐点个数为37个且路径平滑度低不符合实际需求;在空间消耗方面A*算法是本发明算法的7.2倍,在时间耗费方面A*算法是本发明算法的2.5倍。

[0067] 改进后的A*算法和其他路径规划常用算法对比

[0068] 因为实际船舶航行过程中航线的距离和航线拐点的多少决定了航线的好坏,同时规划航线所用的时间也决定了航线是否能及时被使用,所以下面从规划航线的路径长度、拐点个数以及算法所用的时间三个角度,分别将本发明算法与原始A*算法、APF算法和RRT算法进行对比,描述本算法的有益效果。

[0069] 将本发明算法、A*算法、APF算法和RRT算法在相同环境下分别进行路径规划。本发明算法规划的路径如图7所示,A*算法规划出来的路径如图8所示,APF算法规划出来的路径如图9所示,RRT规划出来的路径如图10所示。各种算法的对比情况如表4所示。

[0070] 表4各种算法规划的路径对比

	本算法	A*	APF	RRT
[0071] 路径长度/px	78.79	79.49	79.12	91.10
拐点个数/个	1	37	-	-
所用时间/s	0.031	0.078	39.12	45.92

[0072] 由表4可知,在路径长度方面,本发明算法规划的路径长度最短为78.79像素,A*算法规划的路径长度为79.49像素,APF算法规划的路径长度为79.12像素,RRT算法规划的路径长度为91.10像素。在路径平滑度方面,本发明算法规划的路径拐点个数最少为1个,A*算法、APF算法和RRT算法规划的路径拐点个数远超过本发明算法,其中A*算法和RRT算法规划的路径平滑度极低,不适合船舶航行实际情况。在路径规划的时间耗费方面,本发明算法路径规划所用时间最短,为0.031s,而A*算法路径规划所用时间为0.078s,APF算法路径规划所用时间为39.12s,RRT算法路径规划所用时间为45.92s。

[0073] 上述本发明实施例序号仅仅为了描述,不代表实施例的优劣。

[0074] 在本发明的上述实施例中,对各个实施例的描述都各有侧重,某个实施例中沒有详述的部分,可以参见其他实施例的相关描述。

[0075] 在本申请所提供的几个实施例中,应该理解到,所揭露的技术内容,可通过其它的方式实现。其中,以上所描述的装置实施例仅仅是示意性的,例如所述单元的划分,可以为一种逻辑功能划分,实际实现时可以有另外的划分方式,例如多个单元或组件可以结合或者可以集成到另一个系统,或一些特征可以忽略,或不执行。另一点,所显示或讨论的相互之间的耦合或直接耦合或通信连接可以是通过一些接口,单元或模块的间接耦合或通信连接,可以是电性或其它的形式。

[0076] 所述作为分离部件说明的单元可以是或者也可以不是物理上分开的,作为单元显

示的部件可以是或者也可以不是物理单元,即可以位于一个地方,或者也可以分布到多个单元上。可以根据实际的需要选择其中的部分或者全部单元来实现本实施例方案的目的。

[0077] 另外,在本发明各个实施例中的各功能单元可以集成在一个处理单元中,也可以是各个单元单独物理存在,也可以两个或两个以上单元集成在一个单元中。上述集成的单元既可以采用硬件的形式实现,也可以采用软件功能单元的形式实现。

[0078] 所述集成的单元如果以软件功能单元的形式实现并作为独立的产品销售或使用,可以存储在一个计算机可读取存储介质中。基于这样的理解,本发明的技术方案本质上或者说对现有技术做出贡献的部分或者该技术方案的全部或部分可以以软件产品的形式体现出来,该计算机软件产品存储在一个存储介质中,包括若干指令用以使得一台计算机设备(可为个人计算机、服务器或者网络设备)执行本发明各个实施例所述方法的全部或部分步骤。而前述的存储介质包括:U盘、只读存储器(ROM,Read-Only Memory)、随机存取存储器(RAM,Random Access Memory)、移动硬盘、磁碟或者光盘等各种可以存储程序代码的介质。

[0079] 最后应说明的是:以上各实施例仅用以说明本发明的技术方案,而非对其限制;尽管参照前述各实施例对本发明进行了详细的说明,本领域的普通技术人员应当理解:其依然可以对前述各实施例所记载的技术方案进行修改,或者对其中部分或者全部技术特征进行等同替换;而这些修改或者替换,并不使相应技术方案的本质脱离本发明各实施例技术方案的范围。

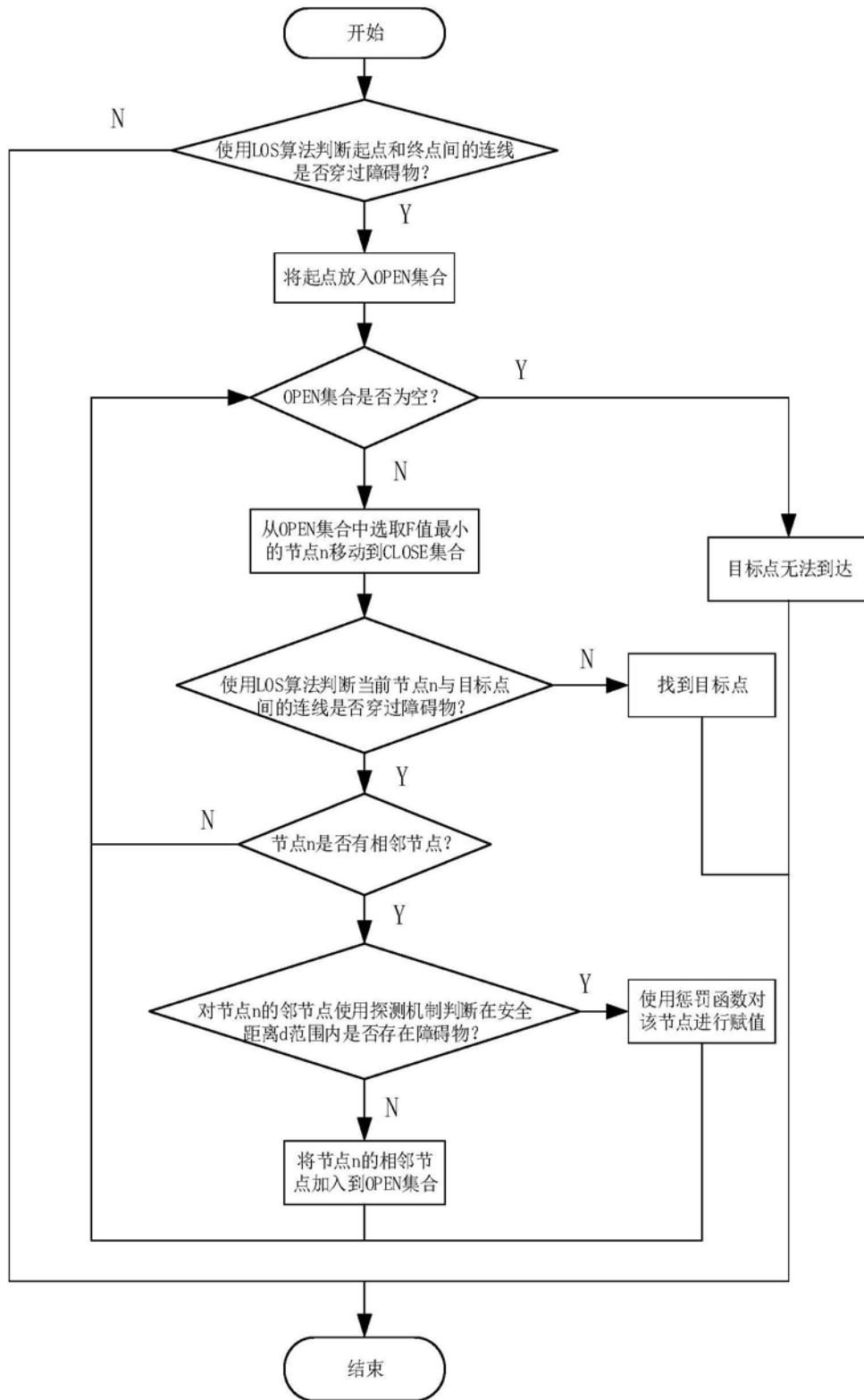


图1

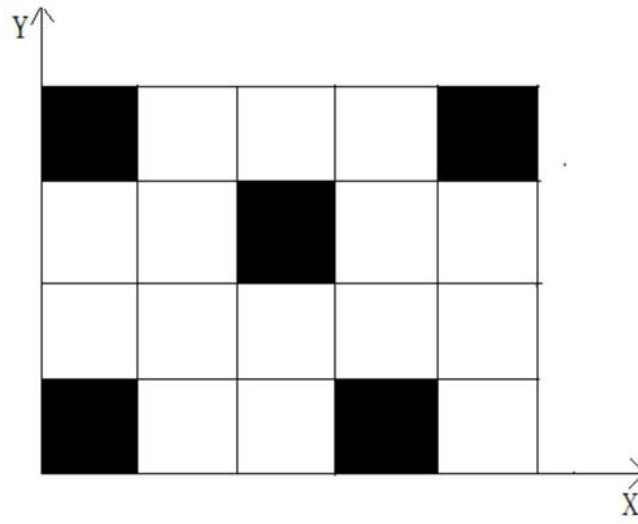


图2

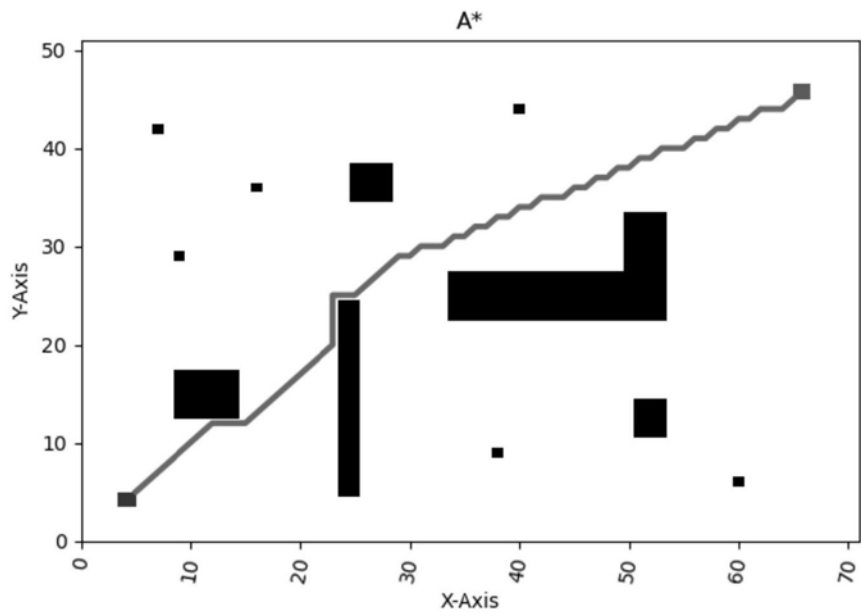


图3

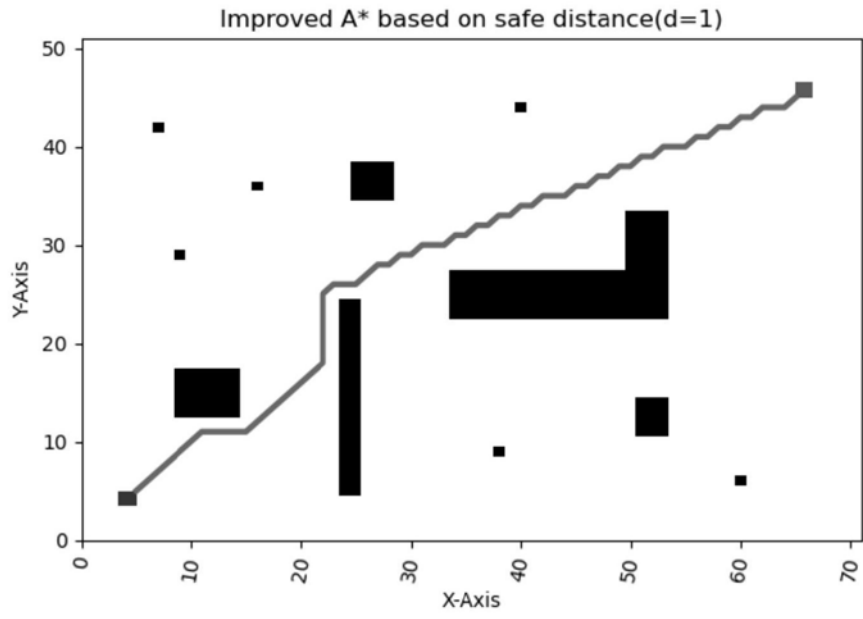


图4

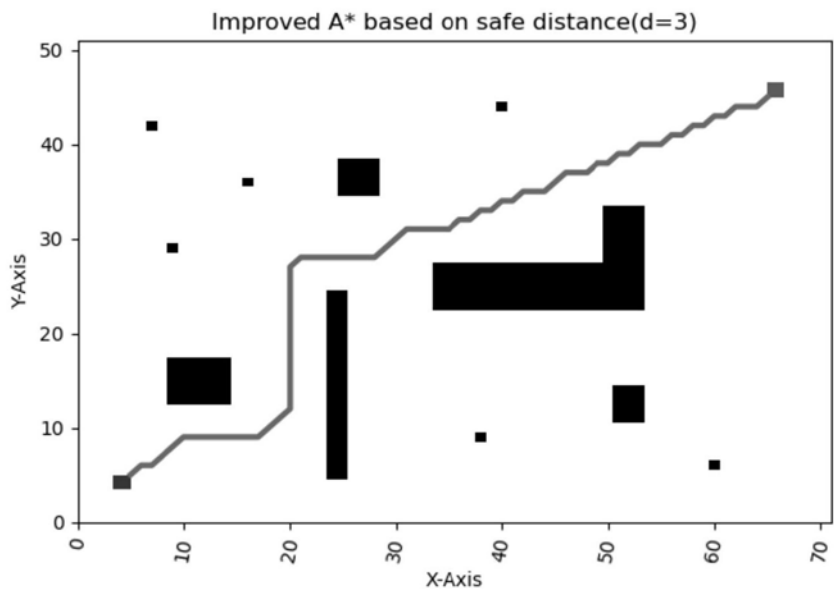


图5

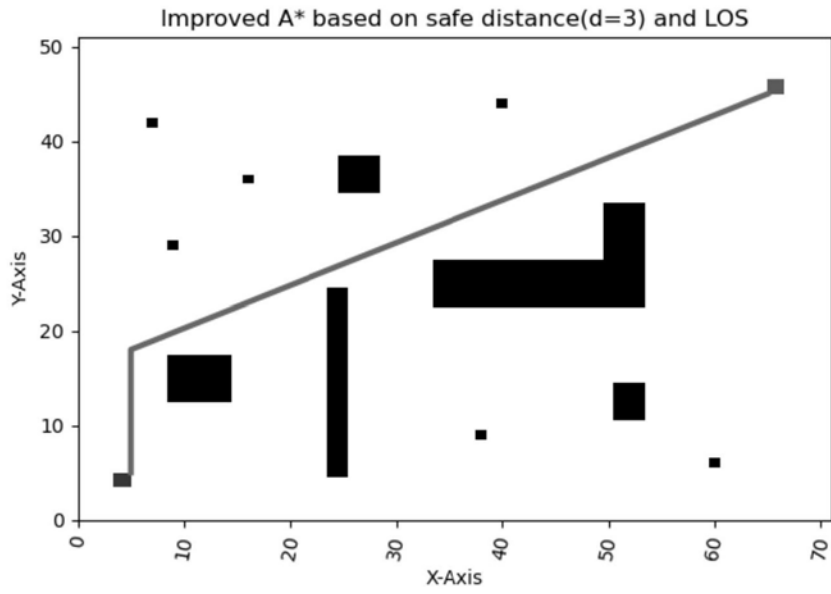


图6

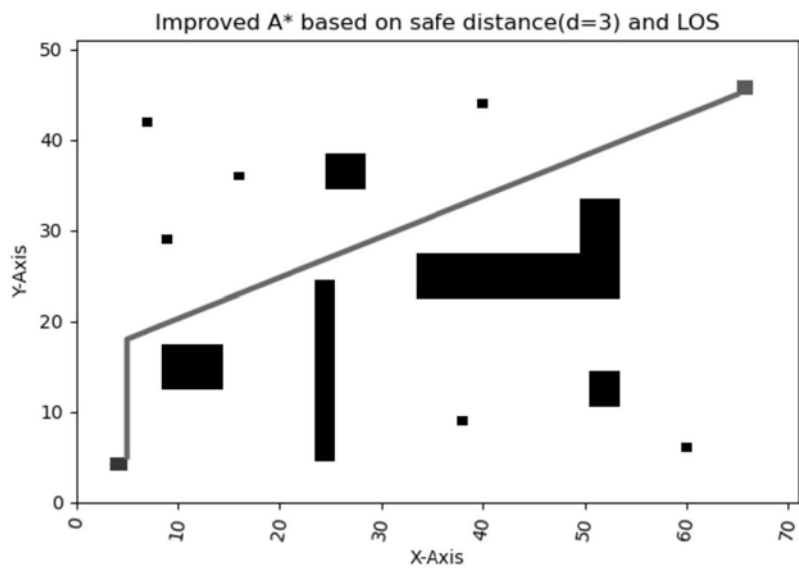


图7

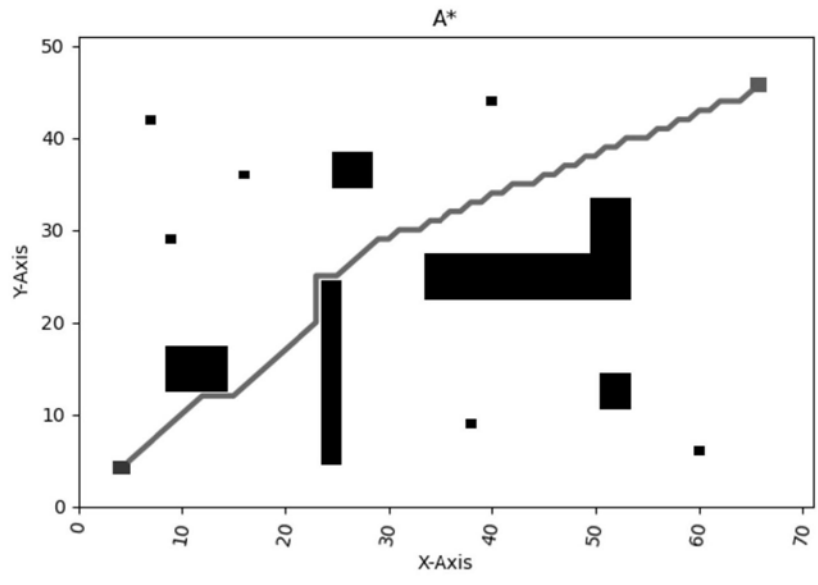


图8

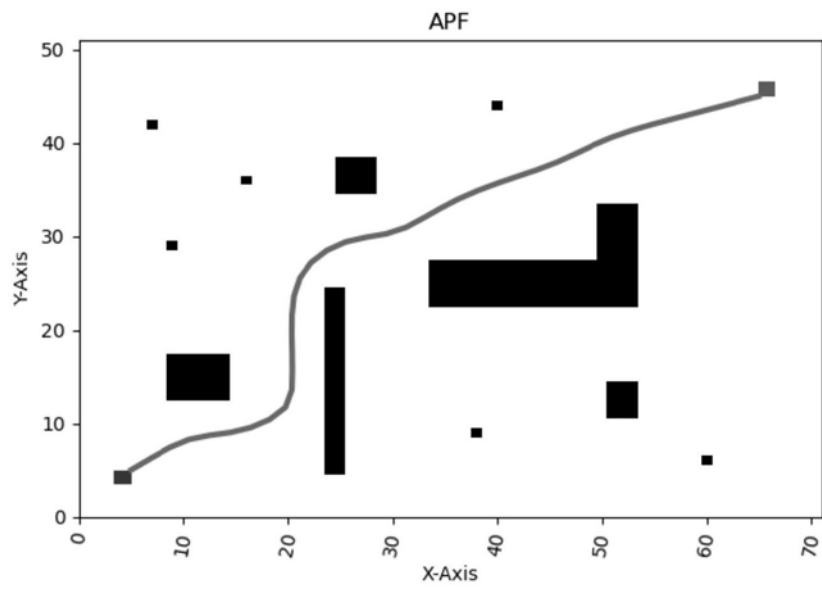


图9

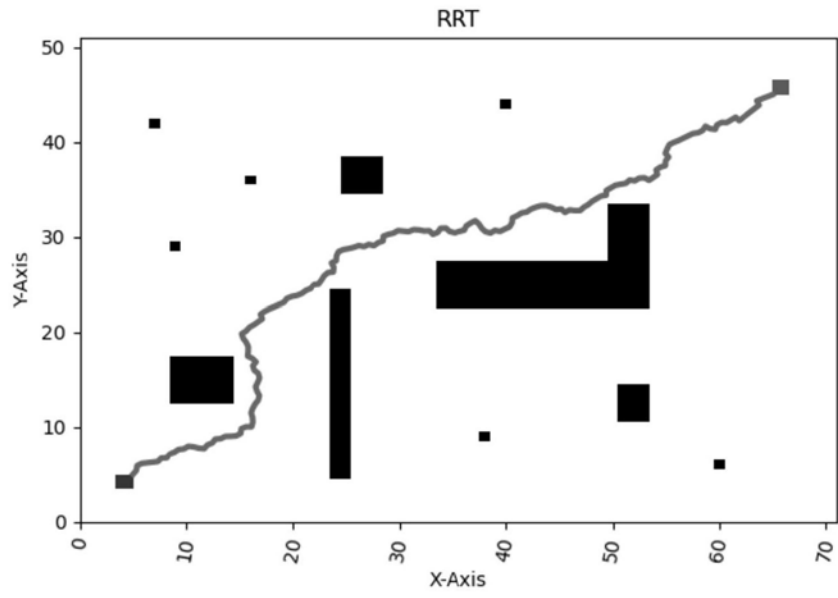


图10