DOI: 10.13382/j.jemi.B2508411

基于改进 Voronoi 骨架图与 DWA 融合的移动机器人路径规划算法*

曾宪阳 梁远生 于 浩 杨红莉
(南京工程学院工程训练中心应用技术学院 南京 211167)

摘 要:在栅格地图环境中,基于 Voronoi 图的路径规划算法具有较好的全局性与完备性,但其生成的路径通常存在转折点过多、冗余路径显著、跟随性较差以及在动态障碍物环境下规划效率偏低等问题。针对上述不足,提出了一种基于改进 Voronoi 骨架图与动态窗口法(dynamic window approach,DWA)融合的路径规划算法(BV-R-GDWA)。该算法首先利用 Voronoi 骨架图的关键点提取与拓扑重构技术,结合障碍物膨胀模型和拐点筛选机制对初始路径进行重规划,从而获得一条在安全距离约束下更短且更平滑的全局优化路径。在局部规划阶段,创新性地设计了动态权重全局路径引导函数,使机器人能够根据当前位置与全局路径偏差自适应调整跟踪策略。实验结果表明,在简单环境中,与 Voronoi 骨架图方法相比,所提全局路径算法在规划时间、路径长度和转折点数量上分别减少了 26.3%、12.9%和 27.3%;在复杂动态环境中,BV-R-GDWA 算法仍能保持较高的规划效率与路径质量,表现出良好的鲁棒性与适应性。主要创新点在于提出了关键点提取与动态权重引导机制,实现了全局路径安全性与局部避障实时性的有效平衡,对提升移动机器人在复杂场景下的导航性能具有重要理论意义和工程应用价值。

关键词:路径规划;Voronoi 图;DWA 算法;Dijkstra 算法
中图分类号:TN964.1 **文献标识码**:A **国家标准学科分类代码**:520.2099

Mobile robot path planning algorithm based on the improved Voronoi skeleton graph and dynamic window approach integration

Zeng Xianyang Liang Yuansheng Yu Hao Yang Hongli
(Engineering Training Center & School of Applied Technology, Nanjing Institute of Technology, Nanjing 211167, China)

Abstract: In grid map environments, path planning algorithms based on Voronoi diagrams offer good globality and completeness. However, the resulting paths often suffer from excessive turning points, significant redundant paths, poor followability, and low planning efficiency in dynamic obstacle environments. To address these shortcomings, this paper proposes a path planning algorithm (BV-R-GDWA) that integrates an improved Voronoi skeleton diagram with the dynamic windowing approach (DWA). This algorithm first utilizes key point extraction and topology reconstruction techniques from the Voronoi skeleton diagram, combined with an obstacle inflation model and inflection point screening mechanism, to replan the initial path, resulting in a shorter and smoother globally optimized path within safety distance constraints. During the local planning phase, this paper innovatively designs a dynamic weighted global path guidance function, enabling the robot to adaptively adjust its tracking strategy based on the deviation between its current position and the global path. Experimental results show that in simple environments, compared with the Voronoi skeleton graph method, the proposed global path algorithm reduces planning time, path length, and the number of turning points by 26.3%, 12.9%, and 27.3%, respectively. In complex dynamic environments, the BV-R-GDWA algorithm can still maintain high planning efficiency and path quality, showing good robustness and adaptability. The main innovation of this paper is the proposed key point extraction and dynamic weight guidance mechanism, which achieves an effective balance between global path safety and local obstacle avoidance real-time performance. It has important theoretical significance and engineering application value for improving the navigation performance of mobile robots in complex scenarios.

Keywords: path planning; Voronoi diagram; DWA algorithm; Dijkstra algorithm

收稿日期:2025-05-26 Received Date: 2025-05-26
*基金项目:国家自然科学基金(11701274)、江苏省自然科学基金(BK20170760)、江苏省研究生科研与实践创新计划(SJCX24_1297,SJCX24_1291)、南京工程学院教学改革重点项目(JXGG2025010, JXGG2025013)资助

0　引　言

随着智能机器人技术和人工智能的快速发展,自主移动机器人在仓储物流、室内服务和复杂环境探索等领域得到了广泛应用。在自主移动机器人研究中,路径规划为自主移动机器人的关键技术之一[1],旨在在已知或未知环境中为机器人寻找一条安全、平滑且高效的可行路径,其性能直接影响机器人导航的效率与安全性。近年来,研究者在全局与局部路径规划算法方面进行了大量研究,形成了多种具有代表性的算法体系。

在全局路径规划方面,A* 算法[2]、Dijkstra 算法、快速扩展随机树(rapidly-exploring random tree, RRT)[3]以及 Voronoi 图算法[4]是目前应用较为广泛的经典方法。A 算法结合了广度优先搜索的完备性和贪心搜索的启发性优点[5],能够在栅格地图中快速生成一条可行路径,但其计算过程需要大量内存,且生成路径拐点较多、靠近障碍物边缘[6]。为提高搜索效率,Harabor 等[7]提出了跳点搜索(jump point search, JPS)算法,通过在跳跃点之间进行搜索以减少中间节点数量,从而提升了搜索速度。Dijkstra 算法[8]作为最短路径问题的经典算法,能在加权图中通过贪心策略逐步扩展节点并记录最短距离,具有良好的完备性,但在大规模地图中计算复杂度较高,导致规划效率下降。

Voronoi 图路径规划算法通过空间分割生成避障路径,能够为机器人移动提供良好的安全裕度[9]。然而,该算法生成的路径通常存在过多转折点和冗余路径,导致路径不够平滑且距离较长。Karavelas 等[10]提出了一种动态 Voronoi 图算法,该算法可以在环境发生变化时实时更新 Voronoi 图,在具有动态障碍物的环境中通过动态调整 Voronoi 图的边界,从而生成适应环境变化的路径,但由于其在删除或更新节点时会引发较大范围的拓扑重构,算法计算量大、实时性不足。由此可见,现有 Voronoi 图类算法虽具有全局完备性与避障优势,但在路径长度、平滑性与计算效率方面仍存在不足。

在局部路径规划方面,典型算法有动态窗口法(dynamic window approach, DWA)与人工势场法(artificial potential field, APF)[11]。DWA 算法基于机器人动力学约束,在速度空间内搜索可行轨迹并实时避障[12],具有良好的响应速度与动态环境适应性,但由于仅依赖局部信息,容易陷入局部最优解。为了提升全局可达性,研究者通常将 DWA 与全局路径规划算法相结合[13-14]。

针对上述问题,本文在 Voronoi 骨架图算法的基础上提出了一种基于 Voronoi 骨架关键点重规划的路径优化算法(BV-R 算法)。该算法对 Voronoi 骨架结构进行关键点提取与连接关系优化,以减少冗余路径和转折点数量,得到初始路径。随后,结合障碍物膨胀模型提取拐点信息,对初始路径进行二次重规划,获得兼顾安全性与平滑性的全局最优路径。为进一步提升算法在动态环境下的导航性能,本文将 BV-R 算法与 DWA 算法相融合,形成 BV-R-GDWA 融合算法,该算法在保持全局路径引导的同时,利用局部规划实现对新增或者动态障碍物的实时规避,从而在保证路径安全与平滑的前提下提升了整体规划效率。

1　Voronoi 骨架图关键点提取与重规划

1.1　Voronoi 骨架图提取

本文使用的是基于 Voronoi 骨架图在二维环境中进行路径规划的一种路径规划方法,首先要先构造 Voronoi 图。二维空间可以描述为 $C \subset R^2$,其中障碍物区域由 $C_{obs} \subset C$ 表示,可供机器人运动的无障碍物区域可以表示为 $C_{free} = C \backslash C_{obs}$[15-16],Voronoi 图使用 C_{obs} 中的点集 $O = \{O_1, O_2, \cdots, O_n\}$ 作为生成点来构造[12],从而得到一种将 C 分区的图结构,其中 $O \subseteq C_{obs}$,$O_i = (x_i, y_i)$ 为二维平面中的某个点[10]。对生成点进行 Delaunay 三角化,得到一个三角形网格,并且每个三角形的外接圆内不包含其他的生成点,三角形圆心坐标为:

$$x_c = \frac{(x_i^2 + y_i^2)(y_j - y_k) + (x_j^2 + y_j^2)(y_k - y_i) + (x_k^2 + y_k^2)(y_i - y_j)}{2(x_i(y_j - y_k) + x_j(y_k - y_i) + x_k(y_i - y_j))} \tag{1}$$

$$y_c = \frac{(x_i^2 + y_i^2)(x_k - x_j) + (x_j^2 + y_j^2)(x_i - x_k) + (x_k^2 + y_k^2)(x_j - x_i)}{2(x_i(y_j - y_k) + x_j(y_k - y_i) + x_k(y_i - y_j))} \tag{2}$$

每个 Delaunay 三角形的外接圆圆心为 Voronoi 图的顶点。每个对象 O_i 都有一个对应的 Voronoi 区域 V_i,这个区域中的所有点都更加接近于 O_i,V_i 中点的集合表示为:

$$V_i = \{v = (x,y) \in C \mid d((x,y), O_i) \leqslant d((x,y), O_j), \forall i \neq j\} \tag{3}$$

所有的 Voronoi 区域 $V_1, V_2, \cdots, V_m$ 组成了广义 Voronoi 图,相邻的 Voronoi 区域共享边界,这个边界上的每个点是到至少两个生成点的距离相等的点,边界点的集合为:

$$V = \{v = (x,y) \in C \mid \exists i \neq j, d((x,y), O_i) = d((x,y), O_j)\} \tag{4}$$

$$d((x,y), O_i) = \sqrt{(x - x_i)^2 + (y - y_i)^2} \tag{5}$$

即 $V = \{v_1, v_2, \cdots, v_n\}$,$n$ 为 Voronoi 边界顶点的数

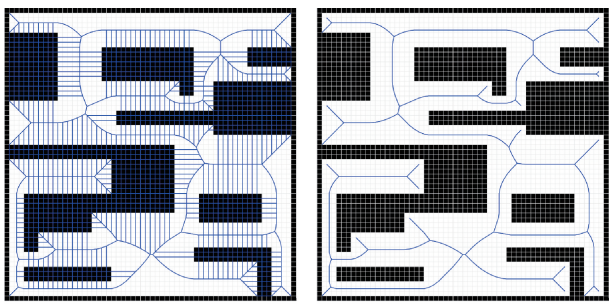
量，对于每个相邻的 v_k, v_{k+1}，都有边 $\{v_k, v_{k+1}\} \in E(k=1, 2, \cdots, m-1)$，$m$ 为 Voronoi 边的数量，$m = n - 1$，此时还要构建邻接矩阵便于判断各个 Voronoi 点之间是否连接，邻接矩阵 $\boldsymbol{A}$ 为一个 $n \times n$ 矩阵，计算方式为：

$$\boldsymbol{A}(i,j) = \begin{cases} d(v_i, v_j), & \{v_i, v_j\} \in E \\ Inf, & \text{其他} \end{cases} \tag{6}$$

式中：$d(v_i, v_j)$ 表示 Voronoi 顶点 v_i 和 v_j 之间的欧氏距离，例如 e_1 就表示顶点 v_1 和 v_2 之间的欧式距离，如果两顶点不能组成 Voronoi 边则两顶点之间的欧氏距离为无穷大，计算得到邻接矩阵为：

$$\boldsymbol{A} = \begin{bmatrix} Inf & e_1 & \cdots & Inf \\ e_1 & Inf & \cdots & Inf \\ \vdots & \vdots & \ddots & Inf \\ Inf & Inf & Inf & Inf \end{bmatrix} \tag{7}$$

由于在栅格地图中，障碍物是由多个栅格组成的，每个栅格都是一个生成点，所以基于这些生成点生成的 Voronoi 边界会穿过障碍物，如图 1(a)所示。



(a) Voronoi图 (b) Voronoi骨架图
(a) Voronoi diagram (b) Voronoi skeleton

图 1 广义 Voronoi 图

Fig. 1 Generalized Voronoi diagram

因此在得到 Voronoi 边界后还要删除在障碍物中的边界，首先对于任意两个相邻的 Voronoi 顶点 $v_i = (x_i, y_i)$，$v_{i+1} = (x_{i+1}, y_{i+1})$ 使用 Bresenham 线算法来生成从 v_i 到 v_{i+1} 的所有离散点，首先计算水平方向和垂直方向的增量：

$$\Delta x = | x_{i+1} - x_i |, \Delta y = | y_{i+1} - y_i | \tag{8}$$

然后计算水平方向和垂直方向的步进方向：

$$s_x = \text{sign}(\Delta x), s_y = \text{sign}(\Delta y)$$

$$\text{sign}(z) = \begin{cases} 1, & z > 0 \\ 0, & z = 0 \\ -1, & z < 0 \end{cases} \tag{9}$$

计算决策变量：

$$err = \frac{\Delta x}{2} \tag{10}$$

如果 $| \Delta x | > | \Delta y |$，则：

$$x = x_i + s_z$$

$$err = err - \Delta y \tag{11}$$

当 $err < 0$ 时，有：

$$y = y + s_y$$

$$err = err + \Delta x \tag{12}$$

直到 $x = x_{i+1}$ 时结束，当 $| \Delta x | \leqslant | \Delta y |$ 步骤相同，只需将 $x_i, \Delta y, \Delta x$ 替换为 $y_i, \Delta x, \Delta y$。最终得到 v_i 到 v_{i+1} 的离散点集 $L = \{(x_j, y_j) \mid j = 1, 2, \cdots, n\}$ 然后判断点集内的点是否位于障碍物区域，如果有点位于障碍物区域则删除对应的边 $\{v_i, v_{i+1}\}$ 并更新邻接矩阵，最终得到 Voronoi 骨架图如图 1(b)所示。

1.2 关键点提取与引导路径规划

得到 Voronoi 骨架图之后再提取 Voronoi 骨架中的关键点，关键点为 Voronoi 骨架图上与 3 个或者多个 Voronoi 边相交的分叉点，可以表示为：

$$B = \left\{ v = (x,y) \in V \middle| \begin{array}{c} \exists i,j,k, i \neq j \neq k, \\ d(v, O_i) = d(v, O_j) = d(v, O_k) \end{array} \right\} \tag{13}$$

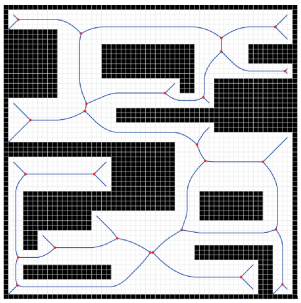
提取出的关键点如图 2(a)所示，提取出 Voronoi 骨架图中的关键点之后还要根据原来 Voronoi 骨架图的连接关系重新连接关键点，重构 Voronoi 骨架图和邻接矩阵，对于有些关键点如果直接连接会穿过障碍物，需要有中间点进行调整，Voronoi 关键点的连接关系确定以及邻接矩阵更新方式如下。

1) 根据邻接矩阵 $\boldsymbol{A}$ 计算每个 Voronoi 点的度数，即计算邻接矩阵 $\boldsymbol{A}$ 中每行或者每列不为 Inf 的元素数量。

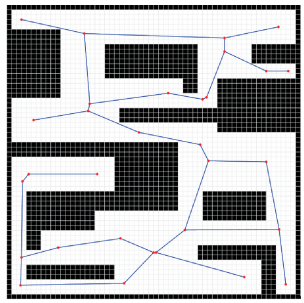
2) 依次遍历所有 Voronoi 点，对于任意一个 Voronoi 点 v_k，若其度数为 2，则找到其邻居节点 v_{k-1}, v_{k+1}，连接点 v_{k-1}, v_{k+1}，如果连线与障碍物碰撞则跳过该点，并将该点作为中间点添加到集合 B 中，如果连线不与障碍物碰撞，计算 $d(v_{k-1}, v_{k+1})$ 填入邻接矩阵 $\boldsymbol{A}$ 中第 $k-1$ 行 $k+1$ 列和第 $k+1$ 行 $k-1$ 列，然后将邻接矩阵 $\boldsymbol{A}$ 的第 k 行和第 k 列元素设置为 Inf；若 v_k 的度数为 1，将邻接矩阵 $\boldsymbol{A}$ 中第 k 行和第 k 列的元素设置为 Inf；若 v_k 的度数 $\geqslant 3$，则跳过该顶点。遍历完所有的 Voronoi 点更新邻接矩阵 $\boldsymbol{A}$。

3) 更新邻接矩阵 $\boldsymbol{A}$，删除邻接矩阵中全为 Inf 的行和列，更新后的邻接矩阵 $\boldsymbol{A}$ 行数或列数与 Voronoi 关键点集合 B 中元素个数相对应。

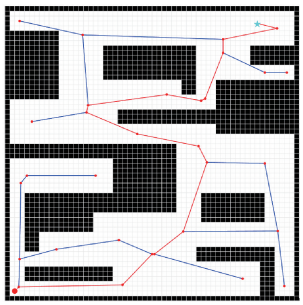
图 2(b)为根据新的邻接矩阵得到的 Voronoi 关键点连接图。相较于图 1(b) Voronoi 骨架图的 439 个顶点，Voronoi 关键点骨架图只有 35 个顶点，极大减小了后续最短路径的计算时间。在得到 Voronoi 关键点连接图之后路径的起点和终点通常不会直接位于 Voronoi 关键点连接图上，本文采用最近原则，将路径的起点与终点直接与最近的不会发生碰撞的 Voronoi 关键点相连，然后基于邻接矩阵使用 Dijkstra 算法计算起点到终点的最短路径，得到引导路径如图 2(c)所示。



(a) Voronoi骨架图关键点
(a) Keypoints on the Voronoi skeleton



(b) Voronoi关键点骨架图
(b) Keypoint-based Voronoi skeleton graph



(c) 引导路径
(c) Guidance path

图 2　Voronoi 骨架图更新与引导路径获取

Fig. 2　Voronoi skeleton graph update and guidance path extraction

1.3　基于障碍物拐点的改进型 Voronoi 全局路径规划

由于引导路径存在很多绕行路线，所以需要对引导路径进行优化。在路径规划中，往往希望得到的全局路径与障碍物保持最小的安全距离 D[17]，即找到的路径 P 要满足：

$$\min_{p\in P} dist(p,C_{obs}) \geqslant D \tag{14}$$

式中：$\min dist(p,C_{obs})$ 表示路径 P 上任意一点 p 到障碍物 C_{obs} 的最小距离。安全距离 D 通常由机器人尺寸、安全裕度、控制误差、传感器精度等因素决定的[18]，D 的大小计算如下：

$$D = R_{robot} + d_{safe} \tag{15}$$

式中：R_{robot} 是机器人外接圆半径；d_{safe} 是额外安全裕度，大小取决于行驶速度、控制精度等。

为了解决安全距离这个问题本文采用膨胀障碍物的方法[19]，通过提取膨胀后障碍物的拐点，使用引导路径对这些拐点筛选，只保留距离引导路径较近的拐点，然后对剩余拐点使用 Dijkstra 算法重新规划，最终得到一条保证安全距离的全局路径。定义障碍物膨胀区域为 C_{obs_D} 该区域由配置空间 C 中到障碍物区域 C_{obs} 的横向和纵向距离分别小于等于安全距离 D 的所有栅格组成的集合，如式(16)所示。

$$C_{obs_D} = \{c \in C \mid d_x(c,C_{obs}) \leqslant D, d_y(c,C_{obs}) \leqslant D\} \tag{16}$$

式中：$d_x(c,C_{obs})$ 表示点 c 到障碍物区域 C_{obs} 的 x 轴方向上的最小距离；$d_y(c,C_{obs})$ 表示点 c 到障碍物区域 C_{obs} 的 y 轴方向上的最小距离。该膨胀操作可以表示为：

$$C_{obs\_D} = C_{obs} \oplus B_D = \bigcup_{o\in C_{obs}} \{c \in C \parallel c_x - o_x \mid \leqslant D \text{ 且 } \mid c_y - o_y \mid \leqslant D\} \tag{17}$$

式中：⊕表示形态学膨胀操作；B_D 表示矩形膨胀核，长宽都为 $2D+1$[10]。

通过上述方法得到膨胀障碍物后的二维栅格地图，该栅格地图可以用二值矩阵 $\boldsymbol{BW}$ 表示：

$$\boldsymbol{BW}(i,j) = \begin{cases} 1, \text{像素属于障碍物} \\ 0, \text{像素属于空闲区域} \end{cases} \tag{18}$$

地图中的障碍物连通区域表示为：

$$W_k = \{(i,j) \in C_k \mid \exists (i',j') \notin C_k, \max(\mid i-i' \mid, \mid j-j' \mid \leqslant 1)\} \tag{19}$$

式中：k 是栅格地图中第 k 个障碍物；C_k 表示第 k 个连通区域的坐标集合。障碍物边界点的集合定义为：

$$W_k = \{(i,j) \in C_k \mid \exists (i',j') \notin C_k, \max(\mid i-i' \mid, \mid j-j' \mid \leqslant 1)\} \tag{20}$$

$$W = \bigcup_{k=1}^{K} W_k \tag{21}$$

式中：W_k 为第 k 个障碍物边界点的集合；W 为地图中所有的障碍物边界点的集合。

在得到障碍物边界点之后删除边界点中共线的点与连续三点之间夹角小于 5°的冗余点，即在障碍物边界点集合 W_k 中对于相邻的 3 个边界点 (x_i,y_i)、(x_{i+1},y_{i+1})、(x_{i+2},y_{i+2})，得到两个向量：

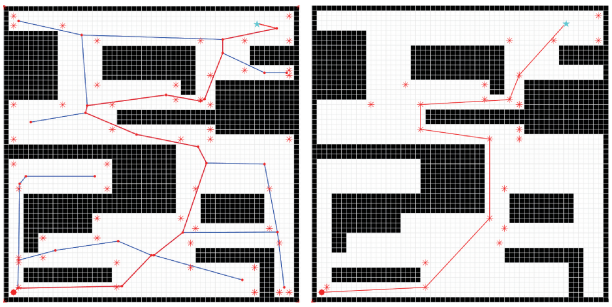
$$\begin{aligned} \overrightarrow{V_1} &= (x_{i+1}-x_i, y_{i+1}-y_i) \\ \overrightarrow{V_2} &= (x_{i+2}-x_{i+1}, y_{i+2}-y_{i+1}) \end{aligned} \tag{22}$$

计算向量之间的夹角：

$$\theta = \arccos\left(\frac{\overrightarrow{V_1}\cdot\overrightarrow{V_2}}{\mid\mid\overrightarrow{V_1}\mid\mid\mid\mid\overrightarrow{V_2}\mid\mid}\right) \tag{23}$$

角度小于 5°则认为点 (x_{i+1},y_{i+1}) 是冗余点，删除该点，依次计算所有的点，剩余的障碍物边界点即视为膨胀后的障碍物拐点如图 3(a)所示。

然后对之前得到的引导路径进行直线插值，通过插值之后的引导路径筛选拐点，只保留距离引导路径较近的拐点，距离阈值的大小与栅格地图和引导路径有关，将引导路径按照拐点分为不同线段，计算所有线段距离障碍物拐点的最小值，取其中最大的值为阈值。得到筛选后的拐点之后再根据膨胀后的障碍物创建新的邻接矩阵，如式(24)所示。



(a) 膨胀后的障碍物拐点 (b) 全局路径
(a) Corner points of inflated obstacles (b) Global path

图 3 全局路径获取

Fig. 3 Global path generation

$$R(p_i,p_j)=\begin{cases}d(p_i,p_j), & i\neq j \text{ and } is\_collision(p_i,p_j)=false\\ Inf, & i=j \text{ or } is\_collision(p_i,p_j)=ture\end{cases} \tag{24}$$

式中：$R(p_i,p_j)$ 为剩余拐点的邻接矩阵，后续用于计算最短路径；$is\_collision(p_i,p_j)=false$ 表示直接连接点 p_i，p_j 不会与膨胀后的障碍物碰撞；$is\_collision(p_i,p_j)=ture$ 表示直接连接点 p_i，p_j 会与膨胀后的障碍物碰撞。基于新的邻接矩阵 $R(p_i,p_j)$ 使用 Dijkstra 算法就能得到与障碍物保持一定安全距离并且路径长度与拐点较少的全局路径 P，最终得到的全局路径如图 3(b)所示。

2 基于动态权重全局引导 DWA 算法(GDWA)

DWA 算法是一种实时的路径规划算法，该算法通过动态窗口限制机器人的速度范围，通过当前环境信息，评估不同轨迹的代价，选择最优轨迹驱动机器人运动[20]，该算法具有较强的实时性和较好的避障能力，但由于受限于局部信息，在路径规划时易陷入局部最优解[21-22]。对此，本文提出了一种全局引导型的 DWA 算法，即 GDWA，在 DWA 算法的评价函数中增加全局路径评价函数，全局路径评价函数的权重设置为动态权重，大小与机器人当前位置与全局路径的距离有关，防止 DWA 算法陷入局部最优解，并且减少 DWA 算法的导航时间。

2.1 移动机器人运动学模型

DWA 算法要依赖机器人运动模型预测机器人在不同控制输入下的轨迹，从而进行路径规划和避障决策[23-24]。机器人底盘模型与运动学模型如图 4 所示。

全局坐标系为二维栅格地图坐标系，机器人整体的线速度 v 和角速度 ω 可以通过两侧轮子的速度 v_r、v_l 计算得到：

$$\begin{aligned}v&=\frac{v_r+v_l}{2}\\ \omega&=\frac{v_r-v_l}{L}\end{aligned} \tag{25}$$

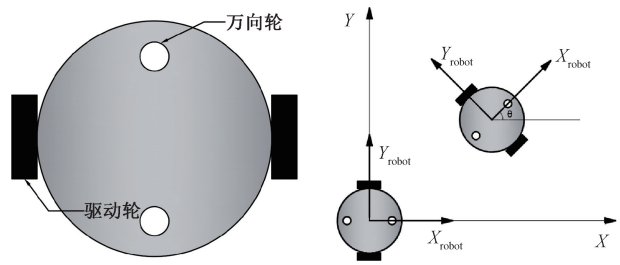


图 4 机器人运动学模型

Fig. 4 Kinematic model of the robot

式中：v_r、v_l 为右轮和左轮的线速度；L 为两侧驱动轮的轴距。

$$\begin{aligned}x_t&=x_{t-1}+v\cos\theta\cdot\Delta t\\ y_t&=y_{t-1}+v\sin\theta\cdot\Delta t\\ \theta_t&=\theta_{t-1}+\omega\cdot\Delta t\end{aligned} \tag{26}$$

式中：x、y 为机器人中心点在全局坐标中的位置；θ 机器人车体的朝向。

2.2 DWA 算法速度窗口采样

在动态窗口法中，速度窗口定义了机器人在未来的短时间内可行的线速度范围和角速度范围[25-26]，线速度 v 和角速度 ω 的取值受到运动学约束、动态约束和环境约束的限制[27-28]。机器人运动学约束如式(27)所示。

$$V_s=\{(v,\omega)\mid v\in[v_{\min},v_{\max}],\omega\in[\omega_{\min},\omega_{\max}]\} \tag{27}$$

式中：$v_{\min}$、$v_{\max}$ 为机器人最小线速度和最大线速度；$\omega_{\min}$、$\omega_{\max}$ 为机器人最小角速度和最大角速度。

在当前线速度 v_c 和角速度 ω_c 的条件下，动态约束如式(28)所示。

$$V_d=\{(v,\omega)\mid v\in[v_c-\dot{v}_{\max}\Delta t,v_c+\dot{v}_{\max}\Delta t],\ \omega\in[\omega_c-\dot{\omega}_{\max}\Delta t,\omega_c+\dot{\omega}_{\max}\Delta t]\} \tag{28}$$

式中：$\dot{v}_{\min}$、$\dot{v}_{\max}$ 为机器人线速度的最小加速度和最大加速度；$\dot{\omega}_{\min}$、$\dot{\omega}_{\max}$ 为机器人角速度的最小加速度与最大加速度。

机器人环境约束如式(29)所示。

$$V_o=\left\{(v,\omega)\mid v\leqslant\sqrt{2d(v,\omega)\dot{v}_{\max}},\omega\leqslant\sqrt{2d(v,\omega)\dot{\omega}_{\max}}\right\} \tag{29}$$

最终机器人在给定时刻的线速度 v 与角速度 ω 的所有可能取值范围可以表示为：

$$V_w=V_s\cap V_d\cap V_o \tag{30}$$

2.3 全局路径引导函数

为了解决 DWA 算法容易陷入局部最优解等问题，本文对 DWA 算法添加了全局路径引导函数，得到 GDWA 算法，全局路径引导函数的大小与 GDWA 算法预

测轨迹终点到全局路径的最小距离有关，该函数的计算方式如式(31)所示。

$$glopath(v,\omega)=\frac{1}{(\min\limits_{(x,y)\in P}\sqrt{(x_{end}-x)^2+(y_{end}-y)^2}+0.1)} \tag{31}$$

式中：(x_{end},y_{end}) 为当前速度 (v,ω) 下预测轨迹的终点坐标；P 为之前得到的全局路径点的集合。GDWA 算法的评价函数如式(32)所示。

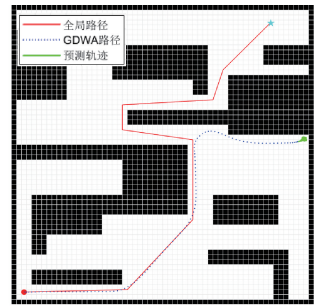
$$G(v,\omega)=\sigma(\alpha\cdot heading(v,\omega)+\beta\cdot distance(v,\omega)+\gamma\cdot velocity(v,\omega)+\eta\cdot glopath(v,\omega)) \tag{32}$$

式中：$heading(v,\omega)$ 为目标追踪代价函数；$distance(v,\omega)$ 为障碍物避让代价函数；$velocity(v,\omega)$ 为速度代价函数；$glopath(v,\omega)$ 为全局路径引导函数；α、β、γ、η 为各评估函数的权重；σ 为归一化操作。

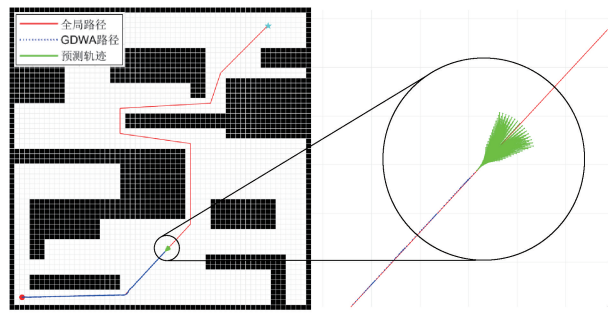
得到全局路径引导函数后还应确定该评价函数的权重，全局路径引导函数的权重对 DWA 算法有着重要的影响，如果全局路径引导函数的权重设置的过小，当机器人在经过地图上的狭长区域即“陷阱区域”时会导致机器人偏离全局路径，再次陷入局部最优解甚至导致路径规划失败，如图 5(a)所示，全局路径引导函数设置的过大会导致机器人在移动时会频繁调整自身位置以减小自身和全局路径的距离，从而导致机器人频繁减速，如图 5(b)所示。

蓝色虚线为每经过 Δt 时间记录机器人当前位置得到的，虚线密集的位置代表机器人在该区域减速，反之虚线稀疏的位置代表机器人在该区域加速。

全局路径引导函数的权重很难用一个定值做到让机器人在遇到“陷阱区域”时即不会偏离全局路径又能减小 DWA 算法的导航时间，因此本文使用动态全局路径引导函数权重，该权重的大小与当前时刻机器人所在位置到全局路径的大小有关，为了减少算法的导航时间，当机器人当前在位置距离全局路径较近时，算法要倾向于选择速度更快的预测轨迹，相对应的全局路径引导函数权重要较小，因此本文设置全局路径引导函数权重最小值在 0.01 左右，当机器人当前位置距离全局路径较远时为了将机器人引导回全局路径，全局路径引导函数权重必需要增大，但是为了保证导航的安全性全局路径引导函数权重要小于避障函数权重，本文的避障函数权重为 0.1，因此设置全局路径引导函数权重最大在 0.07 左右，当机器人偏离全局路径时为了快速将机器人引导回全局路径防止其脱离全局路径，全局路径引导函数权重变化要有较高的敏感程度和响应速度，因此要有较大的系数来控制全局路径引导函数权重变化的速度，并且经过多次实验机器人当前位置与全局路径之间的距离阈值为 0.25 m 时导航所需时间最少，最终得到全局路径引导函



(a) 机器人偏离全局路径
(a) The robot deviates from the global path



(b) 机器人频繁减速
(b) The robot frequently slows down

图 5　全局路径引导函数权重的影响
Fig. 5　The influence of global path guidance function weight

数权重计算公式如式(33)所示。

$$\delta=0.02\cdot\arctan(250\cdot(d(p,r)-0.25))+0.04 \tag{33}$$

式中：$d(p,r)$ 为机器人当前位置到全局路径距离的最小值，计算公式如式(34)所示。

$$d(p,r)=\min\limits_{(x,y)\in P}\sqrt{(x_{robot}-x)^2+(y_{robot}-y)^2} \tag{34}$$

式中：P 表示之前得到的全局路径的点的集合，表示机器人当前位置。当 $d(p,r)$ 从 0 开始增加时 δ 的变化如图 6 所示。

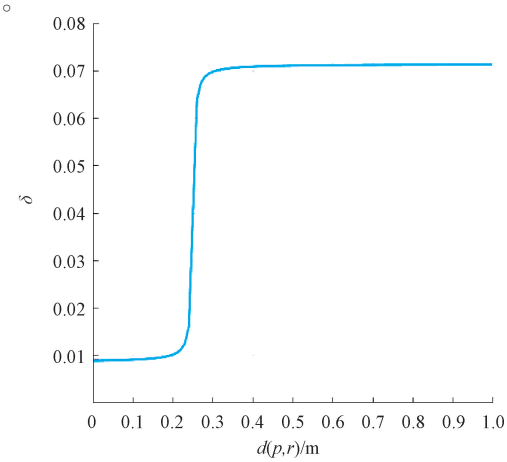


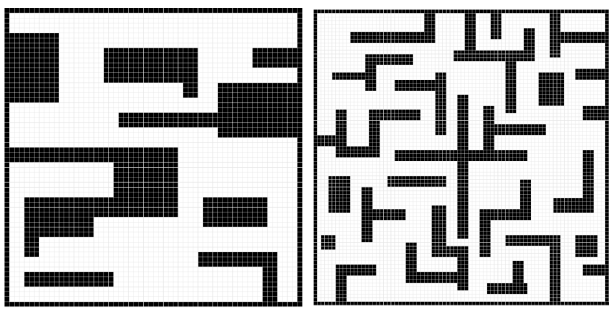
图 6　权重变化曲线
Fig. 6　Weight variation curve

随着机器人当前位置与全局路径之间的距离发生变化,全局路径引导函数权重的下限维持在 0.01 左右,上限维持在 0.07 左右,当 $d(p,r)\leqslant 0.25$ 时全局路径引导函数权重维持在 0.01 附近,防止机器人频繁调整自身与全局路径之间的距离导致的减速,当 $d(p,r)>0.25$ 时全局路径引导函数权重维持在 0.07 附近,防止机器人在遇到“陷阱区域”时偏离全局路径。最后将所有评价函数乘对应权重归一化处理再相加得到综合评价函数,如式(32)所示。

3 实验验证与分析

3.1 仿真实验

为了验证本文方法的可行性和有效性,本文使用计算机软件 MATLABR2023b 进行仿真实验,为了验证算法在不同环复杂程度境下的性能本文设计了两种栅格地图用于实验,一种是 60 m×60 m 的简单栅格地图,简单地图障碍物较小空间开阔,如图 7(a)所示,另一种是 80 m×80 m 的复杂栅格地图,复杂地图内以大密度放置不同规格的障碍物,如图 7(b)所示。



(a) 简单环境 (b)复杂环境
(a) Simple environment (b) Complex environment

图 7 仿真实验地图环境

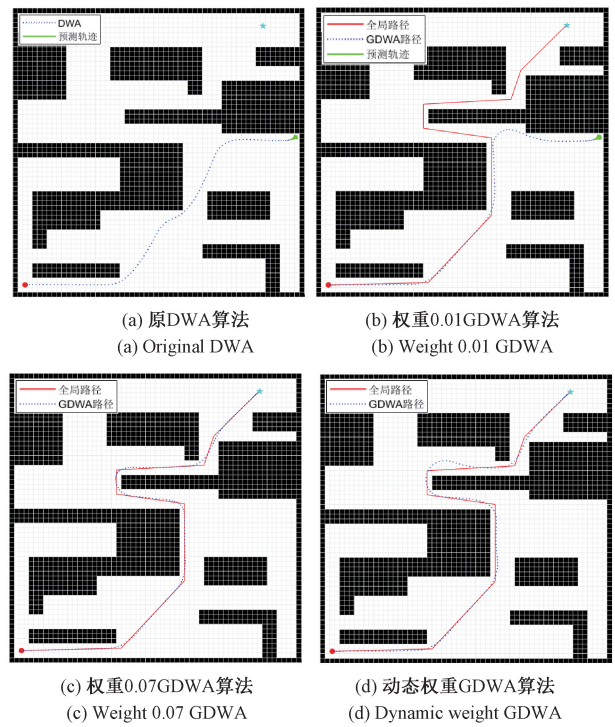
Fig. 7 Simulated experimental map environment

为了验证本文所提的 GDWA 算法的优势,设置了一组对比试验,分别把原 DWA 算法[29]、全局路径引导函数权重为 0.01 的 GDWA 算法(简称权重 0.01GDWA 算法)、全局路径引导函数权重为 0.07 的 GDWA 算法(简称权重 0.07GDWA 算法)与本文的动态全局路径引导函数权重 GDWA 算法(简称动态权重 GDWA 算法)进行对比。为验证动态权重调节机制的有效性,本文选取了全局路径引导函数的两个边界权重值(0.01 和 0.07)作为对比基准。其中,0.01 代表全局引导作用极弱,机器人易偏离全局路径;0.07 代表全局引导作用较强,但易导致频繁减速。这两个权重的选择基于预实验中对算法性能的敏感性分析,旨在通过极端情况的对比,凸显动态权重机制在路径跟踪精度与导航效率之间的平衡优势。DWA 算法的最大线速度为 2 m/s,最大角速度为 0.349 rad/s,最大线加速度为 0.4 m/s^2,最大角加速度为 0.873 rad/s^2。由于较低的目标追踪代价函数权重会使机器人避障时有较大自由度[29],本文 DWA 算法评价函数各项权重分别设置为 $\alpha=0.018$,$\beta=0.1$,$\gamma=0.1$,α 为目标追踪代价函数 $heading(v,\omega)$ 的权重,β 为障碍物避让代价函数 $distance(v,\omega)$ 的权重,γ 为速度代价函数 $velocity(v,\omega)$ 的权重,实验结果如图 8 所示,其中红色实线为全局路径,蓝色虚线为不同全局路径引导函数权重 GDWA 算法得到的路径,绿色实线为算法预测轨迹,表 1 为不同全局路径引导函数权重 GDWA 算法性能指标。

表 1 简单环境不同权重 GDWA 算法实验数据

Table 1 Experimental data of GDWA algorithm with varying guidance weights in a simple environment

算法	路径长度/m	导航时间/s	平均速度/(m·s^{-1})
原 DWA 算法	—	—	—
权重 0.01GDWA 算法	—	—	—
权重 0.07GDWA 算法	107.9	106.8	1.01
动态权重 GDWA 算法	109.5	94.3	1.16



(a) 原DWA算法 (b) 权重0.01GDWA算法
(a) Original DWA (b) Weight 0.01 GDWA

(c) 权重0.07GDWA算法 (d) 动态权重GDWA算法
(c) Weight 0.07 GDWA (d) Dynamic weight GDWA

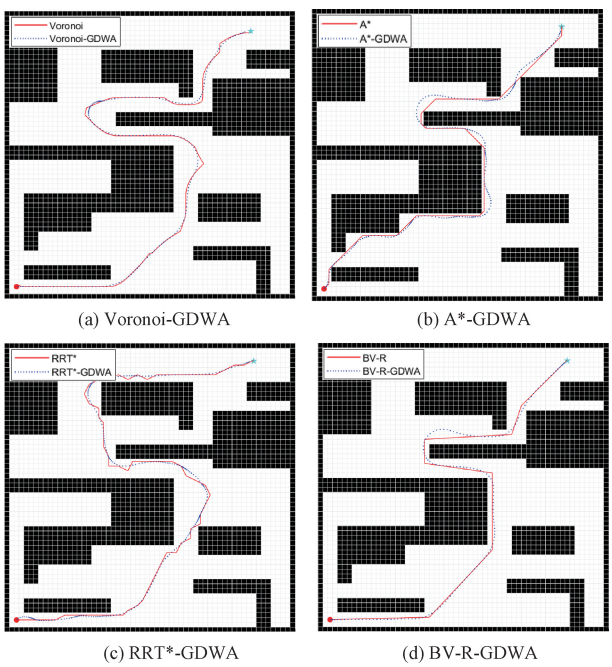
图 8 简单环境不同权重 GDWA 算法对比结果

Fig. 8 Comparison results of GDWA algorithms with different weights in a simple environment

原始 DWA 算法与权重为 0.01 的 GDWA 算法均未能成功到达目标点。在遇到狭窄区域时,这两种算法选

择避让，导致导航失败。相比之下，权重为 0.07 的 GDWA 算法与动态权重 GDWA 算法均成功完成了导航任务。并且两者在路径长度方面差异不大的前提下，动态权重 GDWA 算法相较于权重 0.07 的 GDWA 算法在导航时间上缩短了 13.3%，平均速度提高了 12.9%。

本文选取 A* 算法[2]、RRT* 算法[3] 以及 Voronoi 骨架路径规划算法[4] 作为对比对象，与本文所提出的全局路径规划算法 BV-R 进行性能比较。并且，为验证本文所提出融合算法的优劣，将动态权重 GDWA 算法分别与 3 种经典全局路径规划算法（A*、RRT* 和 Voronoi 骨架方法）进行集成，构建了 3 种融合算法，A*-GDWA、RRT*-GDWA 和 Voronoi-GDWA，并以此作为对比，用于评估本文融合路径规划方法 BV-R-GDWA 的性能。其中 RRT* 算法由于是随机采样，得到的全局路径每次都不相同并且路径规划结果与迭代次数密切相关，实验设置 RRT * 算法的迭代次数为 6 000 次，取 30 次实验的最佳结果作为 RRT* 算法的路径规划结果，本文所使用的小车为差速小车，大小为 200 mm×150 mm，最大速度为 2 m/s，根据式（15）得地图膨胀半径小于 1 m，网格分辨率为 1 m 每格，因此 BV-R 算法计算路径时对障碍物膨胀仅需膨胀一格。简单环境下的路径规划结果如图 9 所示，其中，红色实线表示不同算法所规划的全局路径，蓝色虚线则表示对应融合算法生成的融合路径。表 2 为 4 种全局路径规划算法的性能指标，表 3 为各融合算法的性能评估结果。



(a) Voronoi-GDWA　(b) A*-GDWA

(c) RRT*-GDWA　(d) BV-R-GDWA

图 9　简单环境中不同全局路径融合算法的路径规划结果

Fig. 9　Path planning results of different global path fusion algorithms in a simple environment

从表 2 可以看出在简单环境中，尽管 A* 算法[2] 相比 BV-R 算法所规划的全局路径长度缩短了 2.8%，但其生成的路径与障碍物的最小距离为 0，这在机器人实际跟踪路径时增加了碰撞风险。相比之下，BV-R 算法能够保证路径与障碍物之间最小保持 1 m 的安全距离，从而具有更高的安全性。此外，A* 算法[2] 生成的路径中转折点数量显著多于 BV-R 算法，这导致机器人在跟随路径过程中需要更频繁地减速。因此，尽管 A* [2] 路径在长度上更短，但机器人从起点行驶至目标点的总行驶时间可能长于 BV-R 算法生成的路径所需时间。

表 2　简单环境中全局路径规划算法性能数据

Table 2　Performance data of global path planning algorithms in a simple environment

算法	路径规划时间/s	路径长度/m	与障碍物最小距离/m	拐点数量
Voronoi 骨架图算法[4]	0.38	128.9	1.5	11
A* 算法[2]	0.34	109.1	0	15
RRT* 算法[3]	0.49	136.8	0.5	37
BV-R	0.28	112.3	1	8

从表 3 可以看出 BV-R-GDWA 算法的路径长度相较于 Voronoi-GDWA 算法[4]、A*-GDWA 算法[2] 和 RRT*-GDWA 算法[3] 分别减少了 12.6%、2.2% 和 16.2%；导航时间分别减少了 31.9%、42.8% 和 77.2%；平均速度分别提高了 17.2%、39.8% 和 52.6%。

表 3　简单环境中不同全局路径融合算法实验数据

Table 3　Experimental data of different global path fusion algorithms in a simple environment

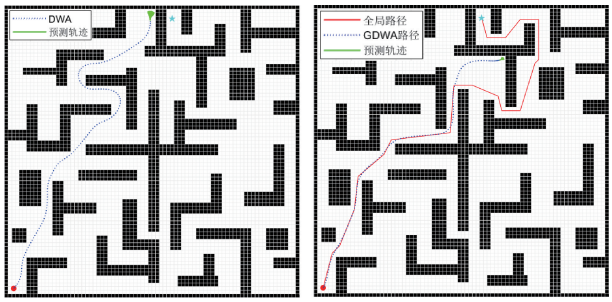
算法	路径长度/m	导航时间/s	平均速度/(m·s^{-1})
Voronoi-GDWA[4]	123.4	124.6	0.99
A*-GDWA[2]	111.9	134.8	0.83
RRT*-GDWA[3]	127.2	167.3	0.76
BV-R-GDWA	109.5	94.4	1.16

在复杂环境中，原 DWA 算法、权重 0.01GDWA 算法、权重 0.07 的 GDWA 算法以及本文提出的动态权重 GDWA 算法的实验结果如图 10 所示。表 4 为不同全局路径引导函数权重 GDWA 算法的性能指标。

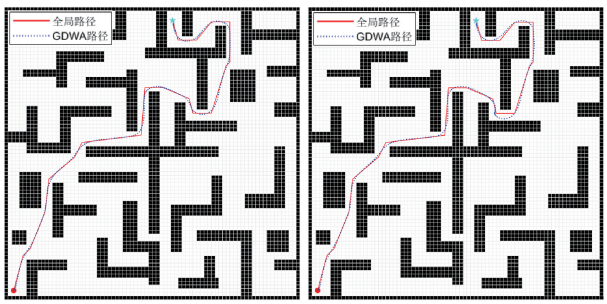
表 4　复杂环境不同权重 GDWA 算法实验数据

Table 4　Experimental data of GDWA algorithm with varying guidance weights in a complex environment

算法	路径长度/m	导航时间/s	平均速度/(m·s^{-1})
原 DWA 算法	—	—	—
权重 0.01GDWA 算法	—	—	—
权重 0.07GDWA 算法	138.2	150.2	0.92
动态权重 GDWA 算法	140.7	130.3	1.08



(a) 原DWA算法 (a) Original DWA　(b) 权重0.01GDWA算法 (b) Weight 0.01 GDWA



(c) 权重0.07GDWA算法 (c) Weight 0.07 GDWA　(d) 动态权重GDWA算法 (d) Dynamic weight GDWA

图 10　复杂环境不同权重 GDWA 算法对比结果

Fig. 10　Comparison results of GDWA algorithm with different guidance weights in a complex environment

在复杂环境中,原始 DWA 算法与权重 0.01GDWA 算法未能完成导航任务,而权重 0.07GDWA 算法与动态权重 GDWA 算法均成功到达目标点。并且,与权重 0.07GDWA 算法相比,动态权重 GDWA 算法在导航时间上减少了 15.3%,平均速度提高了 17.4%。

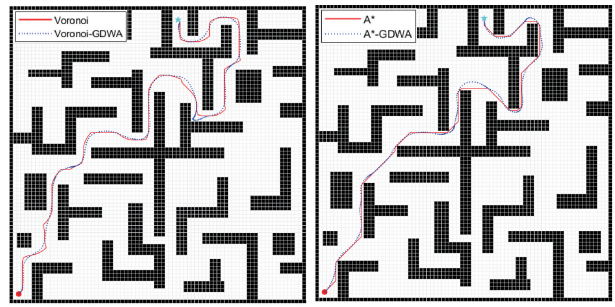
复杂环境下的路径规划结果如图 11 所示。表 5 为 4 种全局路径规划算法在复杂环境中的性能指标,表 6 为相同条件下各融合算法的性能评估结果。

表 5　复杂环境中全局路径规划算法性能数据

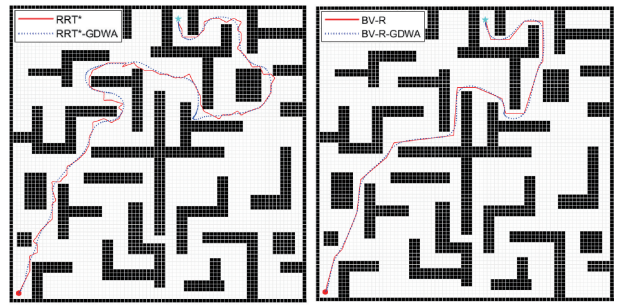
Table 5　Performance data of global path planning algorithms in a complex environment

算法	路径规划时间/s	路径长度/m	与障碍物最小距离/m	拐点数量
Voronoi 骨架图算法[4]	0.56	166.1	1.5	20
A* 算法[2]	0.77	139.5	0	23
RRT* 算法[3]	3.37	183.5	0	32
BV-R	0.43	150.1	1	16

在复杂环境中,与 Voronoi 算法[4]、A* 算法[2]和 RRT* 算法[3]相比,BV-R 算法在路径规划时间上分别减少了 23.2%、44.1% 和 87.2%。在路径长度方面,分别减少了 9.6%、增加了 7.6% 和 减少了 18.2%。路径与障碍物的最小距离分别为 Voronoi 骨架算法[4]为 1.5 m,



(a) Voronoi-GDWA　(b) A*-GDWA



(c) RRT*-GDWA　(d) BV-R-GDWA

图 11　复杂环境中不同全局路径融合算法的路径规划结果

Fig. 11　Path planning results of different global path fusion algorithms in a complex environment

A* 算法[2]为 0 m,RRT* 算法[3]为 0 m,BV-R 算法为 1 m。此外,BV-R 算法生成的路径中路径点数量分别较上述 3 种算法减少了 20.0%、30.4% 和 50.0%。

表 6　复杂环境中不同全局路径融合算法实验数据

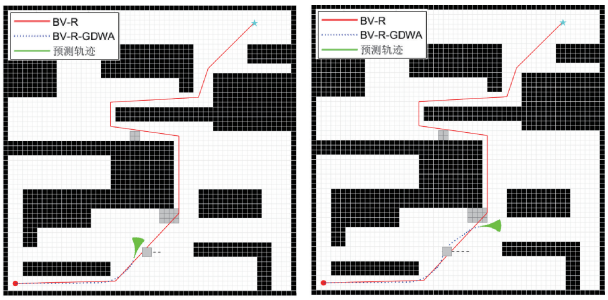
Table 6　Experimental data of different global path fusion algorithms in a complex environment

算法	路径长度/m	导航时间/s	平均速度/(m·s^{-1})
Voronoi-GDWA[4]	159.5	179.2	0.89
A*-GDWA[2]	140.1	166.7	0.84
RRT*-GDWA[3]	198.7	187.4	1.06
BV-R-GDWA	140.7	130.2	1.08

在复杂环境中,BV-R-GDWA 算法的路径长度相比于 Voronoi-GDWA 算法[4]减少了 13.4%,相比 A*-GDWA 算法[2]增加了 0.4%,相比 RRT*-GDWA 算法[3]减少了 41.2%;在导航时间方面,相较于其他 3 种融合算法分别减少了 37.6%、28.1% 和 43.9%;平均速度则分别提高了 17.6%、22.2% 和 1.9%。

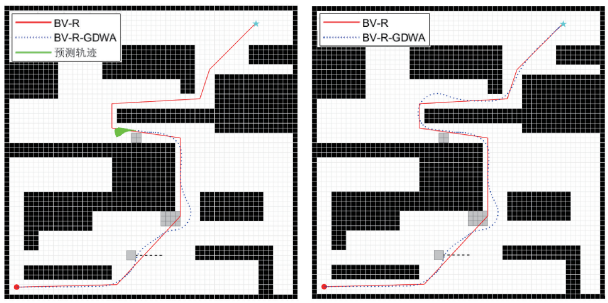
为了验证 BV-R-GDWA 算法对静态、动态新增障碍物的避障能力,本文对简单栅格地图新增了两个静态障碍物和一个动态障碍物对该融合算法进行测试,结果如图 12 所示。

仿真实验结果表明,BV-R-GDWA 算法在导航时具



(a) 正在避开动态未知障碍物
(a) Avoid dynamic unknown obstacles

(b) 正在避开静态未知障碍物1
(b) Avoidance of static unknown obstacle 1



(c) 正在避开静态未知障碍物2
(c) Avoidance of static unknown obstacle 2

(d) 融合算法到达目标点
(d) Reach the goal

图 12　具有未知障碍物场景下融合算法路径规划

Fig. 12　Path planning diagram of the fusion algorithm in an environment with unknown obstacles

有高效性和良好的环境适应性,并且能快速规避动态新增障碍物和静态新增障碍物。

3.2　实车实验

为了进一步验证本文算法在真实场景中的有效性,采用装有激光雷达的小车进行导航避障的实车验证。本文所使用的实物模型为搭载 N10 激光雷达的差速轮小车,两差速论间距为 185 mm,前后有两个万向轮,万向轮间距为 120 mm,如图 13 所示。

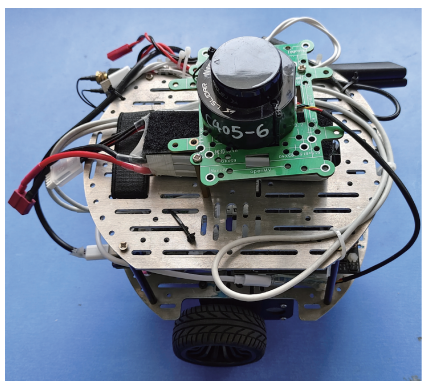
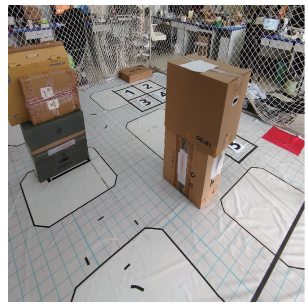


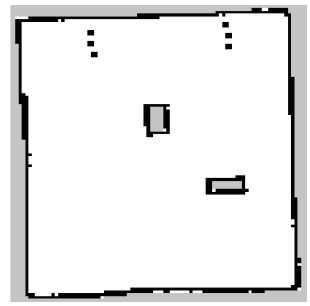
图 13　搭载激光雷达的差速小车

Fig. 13　Differential drive robot equipped with a LiDAR sensor

实验场地如图 14(a)所示,场地面积约为 4 m×4 m。障碍物为不同大小的纸箱,首先要通过激光雷达对试验场地扫描,使用 SLAM(simultaneous localization and mapping)技术创建地图信息,为后续导航做准备,创建好的全局地图如图 14(b)所示。



(a) 实验场地
(a) Experimental environment

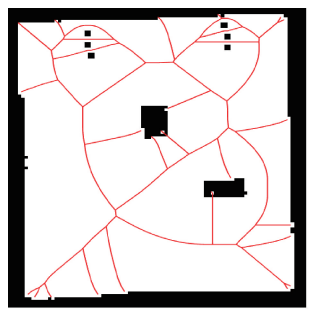


(b) 栅格地图
(b) Grid map

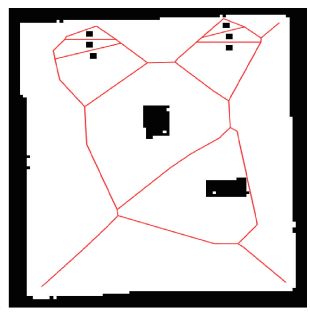
图 14　真实实验场景和环境栅格地图

Fig. 14　Real-world experimental scenario and environment grid map

本文使用的建图算法为 gmapping 建图算法,建好的全局地图为三色地图,为了防止生成的 Voronoi 图位于实验场地之外,还要对全局地图进行二值化处理,并且由于激光雷达等传感器存在误差,全局地图会存在很多噪点,导致生成许多不必要的 Voronoi 边,为了去除地图中的噪点还要对地图进行闭运算,即对地图先进行腐蚀,然后膨胀。最后,原 Voronoi 算法得到的骨架图如图 15(a)所示,本文方法提取到的骨架图如图 15(b)所示。



(a) 原Voronoi骨架图
(a) Original Voronoi skeleton graph

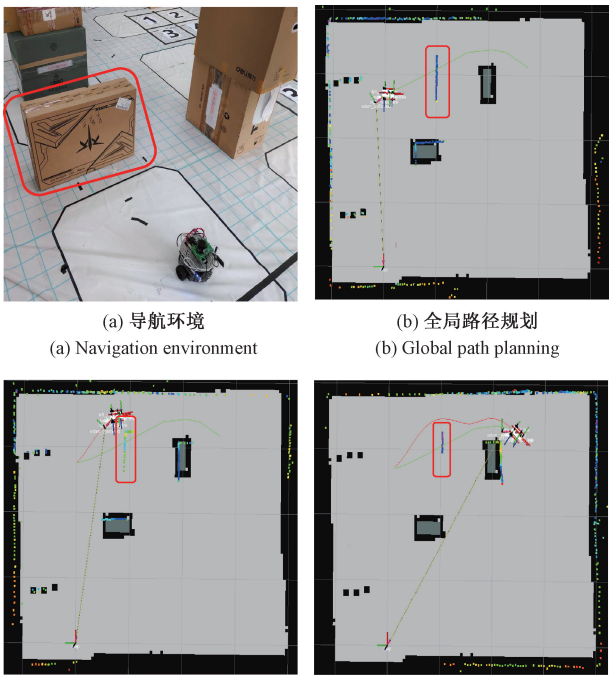


(b) 本文算法骨架图
(b) Skeleton graph of the proposed algorithm

图 15　基于全局地图生成的骨架图

Fig. 15　Skeleton graph generated from the global map

图 16 所示为存在新增障碍物时本文路径规划结果,红色矩形框标注的为新增障碍物,图 16(b)中绿色实线为根据原来地图得到的全局路径,图 16(c)、(d)中红色实线为融合算法得到的路径,从图 16 可以看出,差速小车实际运行轨迹较为平滑,对于新增障碍物可以进行有效规避,能够在实际场景中安全有效运行。



(a) 导航环境
(a) Navigation environment

(b) 全局路径规划
(b) Global path planning

(c) 规避新增障碍物
(c) Avoid newly detected obstacles

(d) 到达目标点
(d) Reach the goal

图 16 差速小车运行轨迹

Fig. 16 Trajectory diagram of the differential drive robot

4 结 论

针对栅格地图中 Voronoi 骨架图路径规划算法生成的路径拐点数量多,生成的路径不适用于机器人跟随且存在较多冗余路径,以及在具有动态障碍物的栅格地图中路径规划效率低下等问题,本文提出了一种基于 Voronoi 骨架图关键点的融合路径规划方法。首先根据栅格地图信息创建 Voronoi 骨架图,提取出 Voronoi 骨架图中的关键点,并根据 Voronoi 骨架图的连接关系构建关键点的邻接矩阵,使用 Dijkstra 算法基于关键点和邻接矩阵生成引导路径,但是初始路径仍有较长绕行距离,本文结合膨胀后的障碍物拐点对引导路径进行重规划,最后得到的全局路径不仅可以与障碍物保持足够的安全距离,并且路径中的转折点较少,路径长度也更短。为了使机器人在跟随路径时能够规避场景中的动-静态障碍物,本文引入了 DWA 算法并与得到的全局路径融合,改进其评价函数,得到一种高效的融合算法 BV-R-GDWA。实验证明本文的融合路径规划算法具有足够的安全性与高效性,能够提升差速小车的环境适应性。

参考文献

[1] 朱洪波,殷宏亮. 分层平滑优化 A* 引导 DWA 用于机器人路径规划[J]. 电子测量与仪器学报,2024,38(9):155-168.
ZHU H B, YIN H L. Hierarchical smooth optimization A* guided DWA for robot path planning [J]. Journal of Electronic Measurement and Instrumentation, 2024, 38(9): 155-168.

[2] HART P E, NILSSON N J, RAPHAEL B. A formal basis for the heuristic determination of minimum cost paths[J]. IEEE transactions on Systems Science and Cybernetics, 1968, 4(2): 100-107.

[3] XIN P, WANG X, LIU X, et al. Improved bidirectional RRT* algorithm for robot path planning[J]. Sensors, 2023, 23(2): 1041.

[4] OGNIEWICZ R L, KÜBLER O. Hierarchic voronoi skeletons[J]. Pattern Recognition, 1995, 28(3): 343-359.

[5] 曾宪阳,张加旺. 改进 A 算法融合 DWA 机器人路径规划研究[J]. 电子测量技术,2025,48(6):20-27.
ZENG X Y, ZHANG J W. Research on integrating DWA robot path planning with improved an algorithm [J]. Electronic Measurement Technology, 2025, 48(6): 20-27.

[6] 马自勇,朱星光,马立东. 改进 A* 和 DWA 的机器人路径规划研究[J]. 现代电子技术,2024,47(20):177-186.
MA Z Y, ZHU X G, MA L D. Research on improving robot path planning with A* and DWA [J]. Modern Electronic Technology, 2024, 47(20): 177-186.

[7] HARABOR D, GRASTIEN A. Improving jump point search[C]. Proceedings of the International Conference on Automated Planning and Scheduling, 2014, 24: 128-135.

[8] 蒋慧灵,方伟,徐天锋,等. 基于 Dijkstra 算法的室内疏散最优路径规划模型[J]. 清华大学学报(自然科学版),2025,65(4):742-749.
JIANG H L, FANG W, XU T F, et al. Optimal path planning model for indoor evacuation based on Dijkstra algorithm [J]. Journal of Tsinghua University (Natural Science Edition), 2025, 65(4): 742-749.

[9] 黄莲花,李光明. 基于 Voronoi 图和快速行进的移动机器人导航路径规划[J]. 机械设计与制造,2023(11):87-92.
HUANG L H, LI G M. Navigation path planning for mobile robots based on Voronoi diagram and rapid travel [J]. Mechanical Design and Manufacturing, 2023 (11): 87-92.

[10] KARAVELAS M I, YVINEC M. Dynamic additively weighted Voronoi diagrams in 2D [C]. European Symposium on Algorithms. Berlin, Heidelberg: Springer Berlin Heidelberg, 2002: 586-598.

[11] ZHANG Z, WU R, PAN Y, et al. A robust reference path selection method for path planning algorithm[J].

IEEE Robotics and Automation Letters, 2022, 7(2): 4837-4844.

[12] 武星,李杨志,臧铁钢,等. 基于 Voronoi 骨架的移动机器人融合路径规划[J]. 机械工程学报,2025,61(5): 165-177.
WU X, LI Y ZH, ZANG T G, et al. Fusion path planning for mobile robots based on Voronoi skeleton [J]. Journal of Mechanical Engineering, 2025, 61 (5): 165-177.

[13] 刘志超,李金凤,王海超. 融合改进 A* 和 DWA 算法的室内机器人路径规划[J]. 制造业自动化,2025,47(2):51-58.
LIU ZH CH, LI J F, WANG H CH. Indoor robot path planning integrating improved A* and DWA algorithms [J]. Manufacturing Automation, 2025, 47 (2): 51-58.

[14] 王勤,魏利胜. 基于改进 HLO 和动态窗口的 AGV 动态避障路径规划算法[J]. 电子测量与仪器学报, 2025, 39(2): 213-221.
WANG Q, WEI L SH. AGV dynamic obstacle avoidance path planning algorithm based on improved HLO and dynamic window [J]. Journal of Electronic Measurement and Instrumentation, 2025, 39 (2): 213-221.

[15] 刘秀婷,高峰. 用于移动机器人全局平滑路径规划的混合遗传算法研究[J]. 机床与液压,2025,53(9): 67-73.
LIU X T, GAO F. Research on hybrid genetic algorithm for global smooth path planning of mobile robots [J]. Machine Tool and Hydraulic, 2025, 53 (9): 67-73.

[16] 万程龙,周焕银,刘凯伦,等. 基于自适应分段步长 RRT-APF 的水下机器人三维路径规划算法[J]. 机床与液压,2025,53(9):31-37.
WAN CH L, ZHOU H Y, LIU K L, et al. Three dimensional path planning algorithm for underwater robots based on adaptive segmented step size RRT-APF [J]. Machine Tool and Hydraulic, 2025, 53 (9): 31-37.

[17] 赵宇新,张志安. 融合改进 A* 与 DWA 的安全移动机器人路径规划[J]. 组合机床与自动化加工技术, 2025(5):37-43.
ZHAO Y X, ZHANG ZH A. Integrating improved A* and DWA for safe mobile robot path planning [J]. Combination Machine Tool and Automation Processing Technology, 2025(5): 37-43.

[18] 付丹丹,李波. 基于改进 RRT* 与 DWA 融合的移动机器人路径规划算法研究[J]. 现代制造工程,2025(4): 91-97.
FU D D, LI B. Research on path planning algorithm for mobile robots based on improved RRT* and DWA fusion [J]. Modern Manufacturing Engineering, 2025(4): 91-97.

[19] 廖功铭,任鸿翔,王德龙,等. 融合改进 A* 与 VO 算法的船舶避碰策略研究[J]. 舰船科学技术, 2025, 47(3):32-38.
LIAO G M, REN H X, WANG D L, et al. Research on ship collision avoidance strategy integrating improved A* and VO algorithms [J]. Journal of Naval Science and Technology, 2025, 47 (3): 32-38.

[20] 樊宏丽,李郁峰,王传松,等. 基于多传感器融合的无人车自主导航系统研究[J]. 激光杂志,2025,46(4): 15-21.
FAN H L, LI Y F, WANG CH S, et al. Research on autonomous navigation system for unmanned vehicles based on multi sensor fusion [J]. Laser Journal, 2025, 46 (4): 15-21.

[21] 谢星星,彭滔,唐灿,等. 面向复杂非结构化三维环境中的 AMR 路径规划[J]. 计算机应用研究,2025, 42(5):1418-1425.
XIE X X, PENG T, TANG C, et al. AMR path planning for complex unstructured 3D environments [J]. Computer Application Research, 2025, 42 (5): 1418-1425.

[22] 赵桂清,崔传辉,高德营,等. 复杂动态环境下先验知识引导机器人路径规划[J]. 现代制造工程,2025(1): 50-56.
ZHAO G Q, CUI CH H, GAO D Y, et al. Prior knowledge guided robot path planning in complex dynamic environments [J]. Modern Manufacturing Engineering, 2025(1): 50-56.

[23] 杨振,李俊丽,杨立炜,等. 安全性 A* 融合 DWA 的分布式多移动机器人路径规划方法[J]. 控制工程, 2024,31(12):2284-2295.
YANG ZH, LI J L, YANG L W, et al. Security A* fusion DWA based distributed multi mobile robot path planning method [J]. Control Engineering, 2024, 31 (12): 2284-2295.

[24] 韩文旭,章翔峰,姜宏,等. 融合人工势场灰狼算法的移动机器人路径规划[J]. 电子测量技术,2025, 48(4):44-50.
HAN W X, ZHANG X F, JIANG H, et al. Path planning for mobile robots integrating artificial potential field grey wolf algorithm [J]. Electronic Measurement Technology, 2025, 48 (4): 44-50.

[25] 宋宇,高岗,梁超,等. 基于多策略改进灰狼算法的无人机路径规划[J]. 电子测量技术,2025,48(1): 84-91.
SONG Y, GAO G, LIANG CH, et al. UAV path planning based on multi strategy improved grey wolf algorithm [J]. Electronic Measurement Technology, 2025, 48 (1): 84-91.

[26] 王敏,石明航,洪梅,等. 基于同步双向 A 星和灰狼优化的多点巡航规划[J]. 电子测量技术,2025,48(1): 1-7.
WANG M, SHI M H, HONG M, et al. Multi point

cruise planning based on synchronous bidirectional A-star and grey wolf optimization [J]. Electronic Measurement Technology, 2025, 48 (1): 1-7.

[27] 冯新,杨雄,曾豫豪. 多场景下基于 IGJO-DWA 算法的机器人路径规划[J]. 传感器与微系统,2024,43(11):131-134,138.
FENG X, YANG X, ZENG Y H. Robot path planning based on IGJO-DWA algorithm in multiple scenarios [J]. Sensors and Microsystems, 2024, 43 (11): 131-134,138.

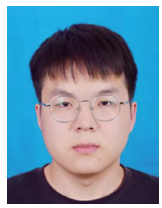
[28] 杨国,吴晓,肖如奇,等. 改进 A* 算法的安全高效室内全局路径规划[J]. 电子测量与仪器学报,2024,38(7):131-142.
YANG G, WU X, XIAO R Q, et al. Improved A* algorithm for safe and efficient indoor global path planning [J]. Journal of Electronic Measurement and Instrumentation, 2024, 38 (7): 131-142.

[29] FOX D, BURGARD W, THRUN S. The dynamic window approach to collision avoidance [J]. IEEE Robotics & Automation Magazine, 1997, 4(1): 23-33.

作者简介

曾宪阳,2003 年于襄樊学院获学士学位,2009 年于浙江师范大学获硕士学位,2018 年于南京大学获博士学位,现为南京工程学院正高级实验师,主要研究方向为智能机器人控制、电工电子技术等。
E-mail: zxymcu@163.com

Zeng Xianyang received his B. Sc. degree from Xiangfan University in 2003, M. Sc. degree from Zhejiang Normal University in 2009, and Ph. D. degree from Nanjing University in 2018. He is now a senior experimentalist (full professor level) at Nanjing Institute of Technology. His main research interests include intelligent robot control, electrical and electronic technology, etc.



梁远生(通信作者),2022 年于南京理工大学紫金学院获得学士学位,现为南京工程学院硕士研究生,主要研究方向为路径规划、SLAM。
Email: 1789353609@qq.com

Liang Yansheng (Corresponding author) received his B. Sc. degree from Zijin College of Nanjing University of Science and Technology in 2022. He is now a M. Sc. candidate at Nanjing Institute of Technology. His main research interests include path planning and SLAM.

杨红莉,2003 年于山西大学获得学士学位,2009 年于南京大学获得博士学位(硕博连读),现为南京工程学院副教授,主要研究方向为数值分析。
E-mail: yanghongli1016@163.com

Yang Hongli received her B. Sc. degree from Shanxi University in 2003, and Ph. D. degree from Nanjing University in 2009 (Combined Master's-Ph. D. Program). She is now an associate professor at Nanjing Institute of Technology. Her main research interest includes numerical analysis.