

基于视觉的室内移动机器人导航避障^①

张增亮, 郑昊天, 刘 爽

(华东理工大学 机械与动力工程学院, 上海 200237)

通信作者: 刘 爽, E-mail: shuangliu@ecust.edu.cn



摘 要: 自主导航与避障作为室内移动机器人完成任务的关键技术, 其重要性日益凸显. 但目前导航的轨迹规划模块生成的轨迹普遍明显偏离原始几何路径, 轨迹偏移严重; 另外, 视觉传感器虽然具有感知信息丰富、成本较低等优势, 但却存在数据量大、处理复杂的问题. 因此本文提出了一种基于视觉的导航避障方法, 使用基于贝塞尔曲线的轨迹规划算法, 用 Bernstein 基表示分段轨迹, 对整个轨迹施加安全约束和动力可行性约束, 有效避免轨迹碰撞, 并改进安全走廊的生成方式, 解决了轨迹偏移严重的问题; 对于室内环境中障碍物的识别与避障, 提出“最近深度法”和“深度比较法”, 排除地面对障碍物识别的干扰, 并使计算资源需求下降. 最后在室内移动机器人平台上进行实验验证, 在不依赖激光雷达硬件的情况下, 只依赖深度相机实现对视野内低矮和悬空、静态和动态障碍物的感知避障, 完成导航任务.

关键词: 轨迹规划; 贝塞尔曲线; 安全走廊; 深度图转激光; 导航; 视觉避障

引用格式: 张增亮, 郑昊天, 刘爽. 基于视觉的室内移动机器人导航避障. 计算机系统应用, 2025, 34(11): 56-67. <http://www.c-s-a.org.cn/1003-3254/10006.html>

Vision-based Navigation and Obstacle Avoidance for Indoor Mobile Robots

ZHANG Zeng-Liang, ZHENG Hao-Tian, LIU Shuang

(School of Mechanical and Power Engineering, East China University of Science and Technology, Shanghai 200237, China)

Abstract: Autonomous navigation and obstacle avoidance are key technologies for indoor mobile robots to complete tasks, and their importance is increasingly emphasized. However, the trajectory planning module in current navigation systems typically generates trajectories that deviate significantly from the original geometric path, resulting in severe trajectory offset. In addition, although visual sensors offer advantages such as rich perceptual information and low cost, they face challenges related to large data volumes and complex processing. This study proposes a visual-based navigation and obstacle avoidance method, utilizing a trajectory planning algorithm based on Bezier curves. Segmented trajectories are represented using Bernstein basis functions, and safety constraints as well as dynamic feasibility constraints are applied to the entire trajectory to effectively avoid collisions. The method for generating safety corridors is also improved to address the issue of severe trajectory deviation. For obstacle recognition and avoidance in indoor environments, the study introduces the “nearest depth method” and the “depth comparison method”, which eliminate interference from ground-level obstacle recognition and reduce computational resource requirements. Finally, experimental verification is conducted on an indoor mobile robot platform, relying solely on a depth camera to perceive and avoid low obstacles within the field of view, without the need for Lidar hardware, thus completing navigation tasks.

Key words: trajectory planning; Bezier curve; safety corridor; depth image to laser scan; navigation; visual obstacle avoidance

① 基金项目: 国家自然科学基金 (51975214)

收稿时间: 2025-04-07; 修改时间: 2025-05-07; 采用时间: 2025-06-09; csa 在线出版时间: 2025-09-30

CNKI 网络首发时间: 2025-10-09

随着科技飞速发展,移动机器人在室内的应用越来越多,可以代替人类从事一些繁琐且重复性的工作,使人类室内生活更加舒适,实现智能家居.轨迹规划是实现移动机器人导航任务的核心模块,直接决定了机器人在执行任务时的运动轨迹,并影响着控制器在跟踪轨迹时的可行性与稳定性.因此,轨迹规划算法的研究一直是移动机器人领域的热点之一^[1].

轨迹规划之前往往需要路径规划,目的是要规划一条从起始点到目标点的无碰撞安全最短路径,完成路径搜索^[2],目前对于室内路径规划方法的研究,常用方法有基于图搜索的方法.基于图搜索的算法主要包括 Dijkstra 算法和 A* 算法等, Dijkstra 算法^[3]虽然可以保证搜索的路径最短,但地图较大时,搜索耗时较长,效率偏低. A* 算法在 Dijkstra 算法的基础上增加了启发式代价函数.郭明阳等^[4]对 A* 算法进行改进,用路径走向约束 A* 算法搜索方向,使路径尽量与路段平行. Hong 等^[5]提出双向 A* 算法,从起始点和目标点同时出发,双向寻找路径,减少搜索空间,搜索效率明显提升.

但是以上路径规划方法往往是纯几何路径规划,规划出的路径不满足移动机器人的运动学约束.对于一些几何转角处需要停下转弯,运动不平滑,不适宜移动机器人跟踪行走,因此需要对路径进行轨迹规划,在路径规划的基础上引入时间参数,对速度和加速度进行规划,强调轨迹连续性、可行性和轨迹参数化,生成满足移动机器人运动学特性的安全且平滑的轨迹^[6,7]. Mellinger 等^[8]提出最小化 snap 算法,采用分段多项式拟合前端路径规划出的几何路径,同时引入时间信息,生成的轨迹具有平滑连续、实时性较强且易引入约束等特点,但是拟合出的轨迹在易碰撞处碰撞风险较大,需要插入多个中继节点,分多段轨迹拟合才能避免碰撞. Liu 等^[9]引入了安全走廊概念,将轨迹生成约束在安全空间内,从而保证生成的轨迹不与障碍物发生碰撞,降低碰撞风险. Lian 等^[10]对最小化 snap 算法进行改进,增加引导功能,解决由于时间分配不均导致的轨迹不完美问题. Zhang 等^[11]提出时空安全走廊,使用多项式的最佳轨迹生成方法,保证轨迹的平滑性与安全性.但是这些算法生成的轨迹普遍存在明显偏离原始几何路径,偏移量大的问题.

此外,移动机器人若要完成导航功能,不仅需要从起始点到目标点的一条轨迹,还需要在跟踪轨迹过程中对障碍物避障.实现避障功能需要构建一张局部地

图,在视觉 SLAM 领域,地图构建通常依赖稀疏特征点或者稠密点云来实现^[12],如 ORB-SLAM2 算法^[13]、VINS-Mono 算法^[14]等.然而这类通过视觉手段构建的地图对于后续导航功能而言,需要经过复杂的坐标转换处理,并不具备直接可用性,且部分信息导航并不需要,有较多冗余的计算处理.但是,视觉可以感知周围三维空间的障碍物信息,这是单线激光只能感受一个平面的信息所不具备的.

综上所述,目前轨迹规划生成的轨迹普遍明显偏离原始几何路径,轨迹偏移量较大,并且视觉数据量大、转换处理复杂.因此本文提出了一种“A*路径规划-贝塞尔轨迹规划-纯视觉避障”的导航避障方法.首先,针对多项式轨迹规划在易碰撞处需要插入多个节点来规避碰撞,且轨迹与原始几何路径偏移严重的问题,提出了一种基于贝塞尔曲线的轨迹规划算法,使用 Bernstein 基替换多项式基,并改进了安全走廊的生成方式,减小轨迹偏移量.其次,针对视觉数据量大、转换处理复杂的问题,提出了“最近深度法”和“深度比较法”,使获取的视觉数据能像激光数据一样简单易处理,实现只依赖深度相机完成视野内障碍物的感知避障.该方法在室内环境有很好的适应性和稳定性,可以很好地完成导航任务.

1 A*路径规划

A* 路径规划算法是图搜索路径规划算法最常用的算法之一^[15],其搜索时依赖一张栅格化的环境地图,在栅格地图中给定起始点和目标点后,搜索出一条连接两点的几何路径,如图 1 所示.

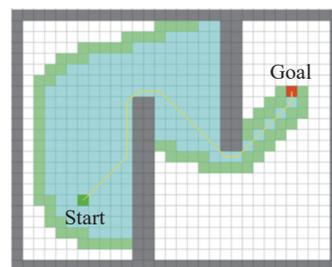


图1 栅格地图及 A* 路径规划图

A* 算法的核心在于代价函数的设计,代价函数公式如下:

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

其中, $g(n)$ 表示从起始节点到当前节点 n 的实际代价

值; $h(n)$ 表示从当前节点 n 到目标节点的预估代价值, 本文采用欧几里得距离计算预估代价值; $f(n)$ 则表示两者的代价值之和. 计算实际代价值时需要明确节点的搜索方式, 常用的有四邻域搜索方式和八邻域搜索方式等. 本文采用八邻域搜索方式(图2), 横向和纵向移动为基础移动代价, 斜向移动为对角线移动代价, 从起始节点移动到当前节点 n 的基础移动代价和对角线移动代价相加可以求得当前节点的实际代价值.

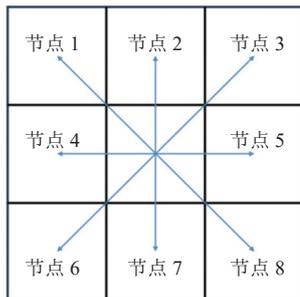


图2 八邻域搜索方式

2 基于贝塞尔曲线的轨迹规划

轨迹规划就是求解一条满足边界约束、连续性约束、安全约束以及动力可行性约束, 且让某一项性能指标达到最优的轨迹^[16]. 本文轨迹规划是在 A* 算法规划出的几何路径的基础上, 满足上述约束后优化生成.

2.1 Bezier 曲线

Bezier 曲线是通过一系列控制点定义的, 控制点的位置直接决定了曲线的形状, Bezier 曲线 $C(t)$ 的定义如下:

$$C(t) = \sum_{i=0}^n p_i B_{n,i}(t), t \in [0, 1] \quad (2)$$

$$B_{n,i}(t) = \frac{n!}{i!(n-i)!} t^i (1-t)^{n-i}, t \in [0, 1] \quad (3)$$

其中, $B_{n,i}(t)$ 称为 Bernstein 基函数, 如果要用贝塞尔曲线替换原始的多项式曲线, 控制点 p_i 被认为是分段轨迹的系数, Bernstein 基对应替换多项式基. Bezier 曲线总是开始于第 1 个控制点, 结束于最后一个控制点, 并且不经过中间控制点, 曲线被完全限制在一系列控制点生成的凸包范围内, 这就是 Bezier 曲线的凸包性质^[17]. 如图3所示, 该性质可以很好地用于轨迹分段以及每一段轨迹安全约束的确定, 同时 Bezier 曲线的导数及多阶导数仍是 Bezier 曲线, 其几何特性不随坐标系的变换而变化, 可以很好地满足轨迹规划对曲线的要求.

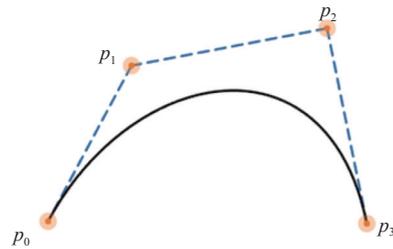


图3 凸包性质

2.2 基于 Bezier 曲线的轨迹表达

对于 Bezier 曲线, 其自变量参数的取值范围是 $[0, 1]$, 而轨迹的时间参数的取值范围是 $[T_{m-1}, T_m]$, 对于每一段轨迹, 都需要一个比例因子 α_m 将时间参数 t 的取值缩放放到 $[0, 1]$, 使两者相对应. 取 $\alpha_m = T_m - T_{m-1}$, 则基于 Bezier 曲线的单维度轨迹表达可表示为式(4).

$$f_\mu = \begin{cases} \alpha_1 \cdot \sum_{i=0}^n p_{\mu 1, i} \cdot B_{n, i} \left(\frac{t - T_0}{\alpha_1} \right), & t \in [T_0, T_1] \\ \alpha_2 \cdot \sum_{i=0}^n p_{\mu 2, i} \cdot B_{n, i} \left(\frac{t - T_1}{\alpha_2} \right), & t \in [T_1, T_2] \\ \vdots & \vdots \\ \alpha_m \cdot \sum_{i=0}^n p_{\mu m, i} \cdot B_{n, i} \left(\frac{t - T_{m-1}}{\alpha_m} \right), & t \in [T_{m-1}, T_m] \end{cases} \quad (4)$$

其中, 总时间为 $T = T_m - T_0$; n 表示 Bezier 曲线的阶数; $\alpha_1, \alpha_2, \dots, \alpha_m$ 为每段 Bezier 曲线的比例因子; μ 表示维度, 对于移动机器人主要是实现二维平面内的轨迹规划, 因此 μ 表示 x 和 y 两个维度, $p_{\mu, i}$ 表示在 μ 维度上第 m 段轨迹的第 i 个控制点; $B_{n, i} \left(\frac{t - T_{m-1}}{\alpha_m} \right)$ 为 Bernstein 基.

轨迹规划的代价函数往往是第 k 阶导数的平方的积分, 由于机器人平台中我们选用深度相机作为主要传感器用于位姿估计, 希望机器人在移动过程中尽可能地减少视差变化率, 这有利于视觉跟踪, 于是 k 的取值为 3, 目标是最小化 jerk. 代价函数可以写为:

$$J = \sum_{\mu=\{x, y\}} \int_0^T \left(\frac{d^k f_\mu(t)}{dt^k} \right)^2 dt \quad (5)$$

由于优化目标函数的积分区间是 $[0, T]$, 需要转化到 $[0, 1]$ 以满足 Bezier 曲线的形式表达. 在单维度上, 第 m 段轨迹记为 $f_{\mu, m}(t)$. 记区间 $[0, 1]$ 上的 Bezier 曲线为 $g_{\mu, m}(\tau)$, 令 $(t - T_{m-1})/\alpha_m = \tau$, 则第 m 段轨迹 μ 维度上的优化目标函数可以写成:

$$\begin{aligned}
 J_{\mu m} &= \int_0^{\alpha_m} \left(\frac{d^k f_{\mu,m}(t)}{dt^k} \right)^2 dt = \int_0^1 \alpha_m \left(\frac{\alpha_m d^k (g_{\mu,m}(\tau))}{d(\alpha_m \tau)^k} \right)^2 d\tau \\
 &= \frac{1}{\alpha_m^{2k-3}} \int_0^1 \left(\frac{d^k g_{\mu,m}(\tau)}{d\tau^k} \right)^2 d\tau
 \end{aligned} \tag{6}$$

优化目标函数确定后,需要确定约束条件,主要有边界约束、连续性约束、安全约束和动力可行性约束这4个约束条件.对于每1条 Bezier 曲线,其高阶导数可以用对应低阶控制点的线性组合表示,同时单维度 μ 上第 m 段轨迹的 l 阶导数仍是 Bezier 曲线形式,则该导数的第 i 项 Bernstein 基的系数为 $s_{\mu m,i}^l$, 导数形式与系数公式如式 (7), 式 (8), 式 (9):

$$f_{\mu m}(t)^{(l)} = \alpha_m^{1-l} \sum_{i=0}^n s_{\mu m,i}^{(l)} B_{n,i} \left(\frac{t - T_{m-1}}{\alpha_m} \right), t \in [T_{m-1}, T_m] \tag{7}$$

$$s_{\mu m,i}^0 = p_{\mu m,i} \tag{8}$$

$$s_{\mu m,i}^{(l)} = \frac{n!}{(n-l)!} (s_{\mu m,i+1}^{(l-1)} - s_{\mu m,i}^{(l-1)}), l \geq 1 \tag{9}$$

边界约束是轨迹的起点应当满足机器人当前的状态,例如机器人当前的位置、朝向、速度、加速度等,同时终点满足目标状态.因此需要在相应控制点上设置相等约束,保证轨迹以相应的状态通过这些点,对于单维度 μ 第 m 段轨迹的开始位置处有明确导数值 $d_{\mu m,start}^{(l)}$, 结束位置处有明确导数值 $d_{\mu m,end}^{(l)}$, 于是有约束:

$$s_{\mu m,0}^{(l)} \alpha_m^{(1-l)} = d_{\mu m,start}^{(l)} \tag{10}$$

$$s_{\mu m,n}^{(l)} \alpha_m^{(1-l)} = d_{\mu m,end}^{(l)} \tag{11}$$

连续性约束是保证曲线 N 阶连续可导, N 往往取决于优化目标函数,如果目标函数中有 N 阶求导,则轨迹至少要 $N+1$ 阶.由于轨迹优化时同时要考虑避障,所以往往不能将一整条几何路径在不分割的情况下完成轨迹优化,通常采用分段的方式实现,因此轨迹的表达式往往是分段函数的形式.于是连续性约束就需要保证分段函数轨迹在分段位置处上一段轨迹的终点与下一段轨迹的起点 N 阶连续,即位置、速度、加速度等连续.连续性约束通过在两条连续曲线的对应控制点之间设置相等约束来实现.对于第 m 段和 $m+1$ 段曲线,有约束如下:

$$s_{\mu m,n}^{(l)} \alpha_m^{(1-l)} = s_{\mu(m+1),0}^{(l)} \alpha_{m+1}^{(1-l)}, s_{\mu m,i}^{(0)} = p_{\mu m,i} \tag{12}$$

安全约束是保证优化后的轨迹在自由安全的空间内,确保机器人不与环境中的障碍物发生碰撞.由于 Bezier 曲线的凸包性质,对于每一段轨迹,只需保证所有控制点都在安全约束形成的凸多边形内,那么这些控制点构成的凸包肯定在凸多边形内,意味着整个轨迹段也被限制在凸多边形内,从而保证轨迹不会与障碍物发生碰撞.通过在控制点上增加边界限制来施加安全约束,记 $\beta_{\mu m}^-$ 和 $\beta_{\mu m}^+$ 分别为每段轨迹安全矩形的上下边界,一系列安全矩形构成安全走廊,安全走廊的生成方式如图 4. 所以对于第 m 段控制点有约束如下:

$$\beta_{\mu m}^- \leq p_{\mu m,i} \leq \beta_{\mu m}^+, i = 0, 1, 2, \dots, n \tag{13}$$

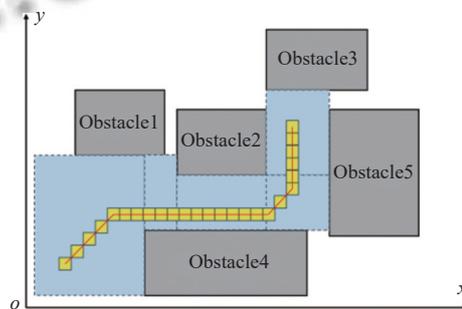


图 4 原始安全走廊生成方式

动力可行性约束是保证优化后轨迹的速度和加速度限制在机器人实际能达到的范围内,保证机器人能正常跟踪轨迹, v 表示速度, a 表示加速度,有约束如下:

$$\begin{cases} v_{\max}^- \leq n(p_{\mu m,i} - p_{\mu m,i-1}) \leq v_{\max}^+ \\ a_{\max}^- \leq n(n-1)(p_{\mu m,i} - 2p_{\mu m,i-1} + p_{\mu m,i-2})/\alpha_m \leq a_{\max}^+ \end{cases} \tag{14}$$

将控制点向量记为 P , 将优化目标函数写为二次型的形式 $P^T Q P$, 其中 Q 为目标函数的 Hessian 矩阵. 将轨迹规划的边界约束和连续性约束表述为线性等式约束 ($A_{eq} P = B_{eq}$), 将安全约束和动力可行性约束表述为线性不等式约束 ($A_{ie} P \leq B_{ie}$), 其中 $P = [P_0, P_1, \dots, P_n]$. 于是将轨迹规划问题转化为一个凸二次规划 (QP) 问题, 如式 (15), 进行二次规划求解, 便可以求得每一段贝塞尔轨迹的控制点, 获得优化后的轨迹.

$$\begin{cases} \min & P^T Q P \\ \text{s.t.} & \begin{cases} A_{eq} P = B_{eq} \\ A_{ie} P \leq B_{ie} \end{cases} \end{cases} \tag{15}$$

2.3 改进安全走廊的生成

安全走廊是根据前端几何路径膨胀生成的一个凸

多边形, 用作轨迹生成时的安全约束, 为了简化安全约束的表达, 安全走廊往往是由一系列安全矩形合并构成, 每一个矩形框中对应一段分段轨迹. 原始安全走廊的生成方式见图4, 黄色栅格点为几何路径的路径点, 红色线条为几何路径, 蓝色区域为根据路径点膨胀生成的矩形, 对于路径中的每个节点, 先确定节点位置在 x 、 y 两个方向上到最近障碍物的距离 D_{x_max} 和 D_{y_max} , 进而确定最大自由面积, 但可能会出现矩形包含的情况, 这是不必要的, 因此需要修剪掉走廊中重复的矩形^[18].

原始安全走廊的生成方式会导致生成的轨迹偏离前端路径规划搜索出的几何路径, 在空闲区域较大时, 轨迹偏移现象尤为严重, 本文对安全走廊的生成方式进行了改进, 如图5. 针对前端 A* 路径搜索出的几何路径, 提出了更具针对性的安全走廊生成策略, 根据第1节介绍的 A* 的八邻域搜索方式, 得到的几何路径基本只由3类线段构成: 横线、竖线和斜线, 因此针对每一类使用对应的膨胀矩形生成方式.

(1) 横线: 首先确定 x 方向上横线的长度, 然后 x 方向上横线两端向外膨胀 R_{range} , y 方向上向外膨胀 R_{range} , 膨胀时保证不与障碍物发生碰撞, 其中 R_{range} 为膨胀半径.

(2) 竖线: 首先确定 y 方向上横线的长度, 然后 y 方向上竖线两端向外膨胀 R_{range} , x 方向上向外膨胀 R_{range} .

(3) 斜线: 首先确定以该斜线为对角线的矩形, x 与 y 方向上都向外膨胀 R_{range} . 如果膨胀过程中与障碍物发生碰撞, 则将大矩形在碰撞处分割成两个小矩形, 即再添加一段分段轨迹, 从而对碰撞进行规避.

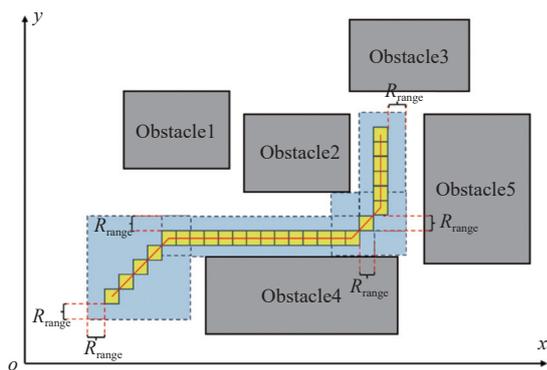


图5 改进安全走廊生成方式

3 基于深度图构建局部地图实现避障

由于通过视觉手段构建的稀疏特征点或者稠密点云地图无法直接用于后续导航功能, 需要经过复杂的

转换和处理. 相较而言, 激光雷达 SLAM 常用的二维概率栅格地图的构建方式更为简便, 故希望利用相机数据构建地图时能够在不丢失三维信息的同时又能让相机数据像激光数据一样简单易于处理. 于是本文提出了一种利用深度图将视觉信息转化成激光信息构建局部地图实现避障的方法.

3.1 最近深度法

构建局部地图时可以只考虑距离机器人最近的障碍物, 远处的障碍物可以等走近后再处理, 本文提出了一种“最近深度法”, 实现将三维的深度信息转换为二维的激光距离信息, 可以有效避免复杂的深度点云处理. 最近深度法的原理如图6, 选取整张深度图中的每一列, 记第 j 列中第 i 行的有效深度值为 $D_{(i,j)}$, 记第 j 列中最小的有效深度值为 min_D_j , 将每一列最近的有效深度值 min_D_j 作为该列的深度值 D_j , 从而实现转换为单线激光深度扫描.

$$min_D_j = \min\{D_{(0,j)}, D_{(1,j)}, \dots, D_{(i,j)}\} \quad (16)$$

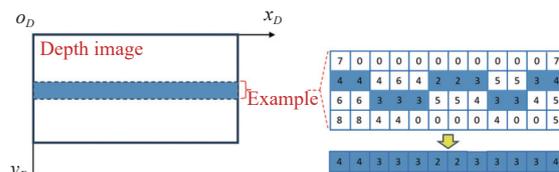


图6 最近深度法

深度图转换激光信息如图7所示, 记相机坐标系下实际空间中的点 p_C 在深度图像的像素坐标为 p_D , 相机光心在像素坐标系下的 x 方向偏移量为 c_x , 相机焦距为 f_x . 对于任意一个点 p_C , 以中心射线往左为正、往右为负, 可以根据像素坐标 p_D 计算出其相对中心射线的夹角 θ_j :

$$\theta_j = -\arctan\left(\frac{j - c_x}{f_x}\right) \quad (17)$$

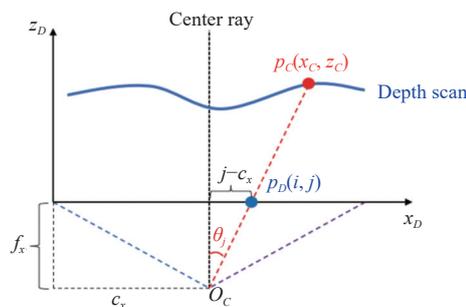


图7 深度图转换激光扫描示意图

取深度图每列最小的有效深度值 \min_D_j 作为该列的激光扫描深度, 角度为上文计算的每一列相对扫描中心射线的夹角 θ_j , 最终实现将整个深度图深度数据压成一条激光数据 $\{\min_D_j, \theta_j\}$.

3.2 深度比较法

采用最近深度法直接将整张深度图转换为激光信息会存在两个问题: 一是若相机摆放位置较低或者相机有朝下的俯仰角, 则深度图中会有地面信息. 由于地面也是有深度的, 直接用最近深度法会将地面误认为障碍物信息, 无法区分低矮障碍物与地面, 这是不符合预期的, 如图 8(a) 所示. 另一个是若相机放置位置较高或相机有朝上的俯仰角, 则深度图中会有悬空物体信息, 比如门框. 悬空物体高度远高于机器人高度, 是不影响机器人正常通行的, 而用最近深度法也会将其误认为障碍物信息, 同样是不符合预期的, 如图 8(b) 所示, 但这两个问题本质上属于同一类问题.

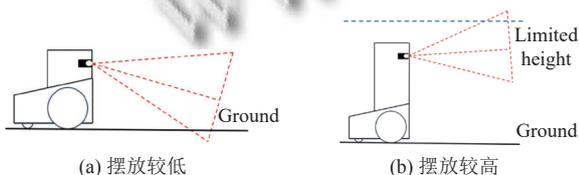


图 8 相机摆放高度与角度示意图

针对该问题, 本文提出了“深度比较法”, 首先针对低矮障碍物问题, 需要根据相机安装高度与俯仰角, 计算出每一行列像素点对应的地面的深度值, 深度观测示意图如图 9. 记相机光心距离 O_C , 地面高度为 C_{Height} , 相机与水平面俯仰角为 α (逆时针为正), 相机焦距与光心在像素坐标系下的 y 偏移量为 c_y , 相机 y 焦距为 f_y , 设地面安全距离为 d , 则安全距离 d 在深度图中投影的高度 u_d , 记深度图的第 $c_y + u_d$ 行, 第中间列地面对应的投影深度值 $R_{c_y + u_d}$, 第 j 列 (其与中间列的夹角为 β_j) 地面对应的投影深度值 $L_{(c_y + u_d, j)}$. 根据图 9(a)、(b) 两图, 根据相似三角形和图形几何分析, 可以得到式 (18), 并最终求得深度图的第 $c_y + u_d$ 行, 第 j 列地面对应的深度值 $D_{(c_y + u_d, j)}$.

$$\begin{cases} d = C_{\text{Height}} / \cos(\alpha) - R_{c_y + u_d} \tan(\alpha) \\ f_y / R_{c_y + u_d} = u_d / d \\ R_{c_y + u_d} = C_{\text{Height}} f_y / (u_d \cos(\alpha) + f_y \sin(\alpha)) \\ L_{(c_y + u_d, j)} = R_{c_y + u_d} / \cos(\beta_j) \\ D_{(c_y + u_d, j)} = \sqrt{L_{(c_y + u_d, j)}^2 + d^2} \end{cases} \quad (18)$$

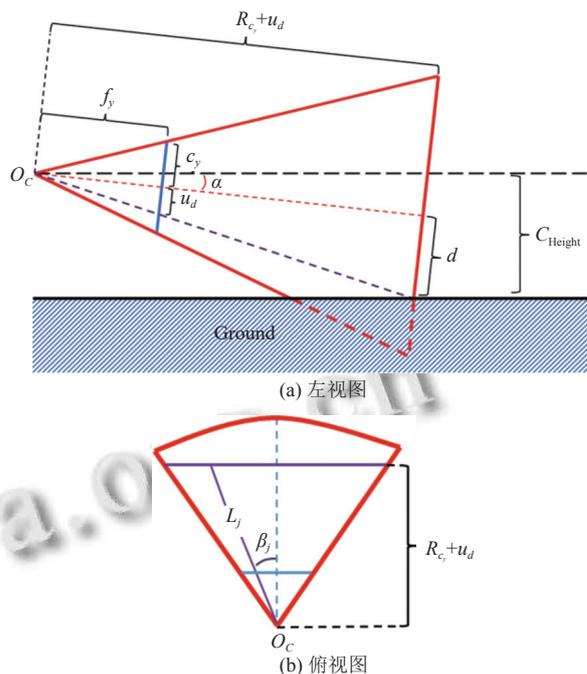


图 9 深度观测示意图

根据计算出每一行列像素点对应的地面深度值 $D_{i,j}$, 可以生成一张“深度标定图”. 深度标定图中的每一个像素点对应相机在此安装角度下地面的深度值. 将机器人在运动过程中获取的实际深度图与深度标定图中每个像素点作对比, 若像素点处实际获取的深度值 $d_{i,j} < D_{i,j}$, 则认为该像素点处存在障碍物; 若 $d_{i,j} \geq D_{i,j}$, 则认为“打空”, 没有障碍物. 由此便可区分低矮障碍物与地面, 排除地面对识别低矮障碍物的干扰.

对于悬空物体, 加入高度 1.8 m 的限制条件, 即对于高于 1.8 m 的悬空物体认为不是障碍物, 由于悬空物体的处理方法与地面是一样的, 这里不再详细介绍.

3.3 局部地图更新全局地图实现避障

当深度图转换成单线激光扫描数据, 便可以进行二维概率栅格局部地图的构建, 其构建的核心是二值贝叶斯原理, 如果每个栅格存在障碍物的置信度大于占据阈值, 则认为该栅格存在障碍物, 如果置信度小于空闲阈值, 则认为该栅格是空闲可通行的. 构建好机器人当前位置的局部地图之后, 需要将局部地图更新到最原始的全局地图上, 其中涉及两者的坐标转换, 局部地图坐标系与机器人坐标系往往是一致的^[19]. 在二维平面上, 全局坐标系到局部坐标系的转换矩阵包括平移矩阵 T_g 和旋转矩阵 R_g , 如式 (19)、式 (20) 所示.

$$T_g = [x_g, y_g]^T \tag{19}$$

$$R_g = \begin{bmatrix} \cos \varphi & \sin \varphi \\ -\sin \varphi & \cos \varphi \end{bmatrix} \tag{20}$$

其中, φ 为机器人的偏航角, 若将全局坐标系中的点 G_i 转换到局部坐标系, 那么它对应局部坐标系下的坐标 L_i 为:

$$L_i = R_g G_i + T_g \tag{21}$$

由于需要将局部地图更新到全局地图中, 所以将局部坐标系中的点 L_i 对应转换到全局坐标系下, 如式 (22) 所示, 据此可以实现局部地图对全局地图更新.

$$G_i = R_g^{-1} (L_i - T_g) \tag{22}$$

图 10 描述了本文提出的一种基于视觉的室内移动机器人导航避障方法.

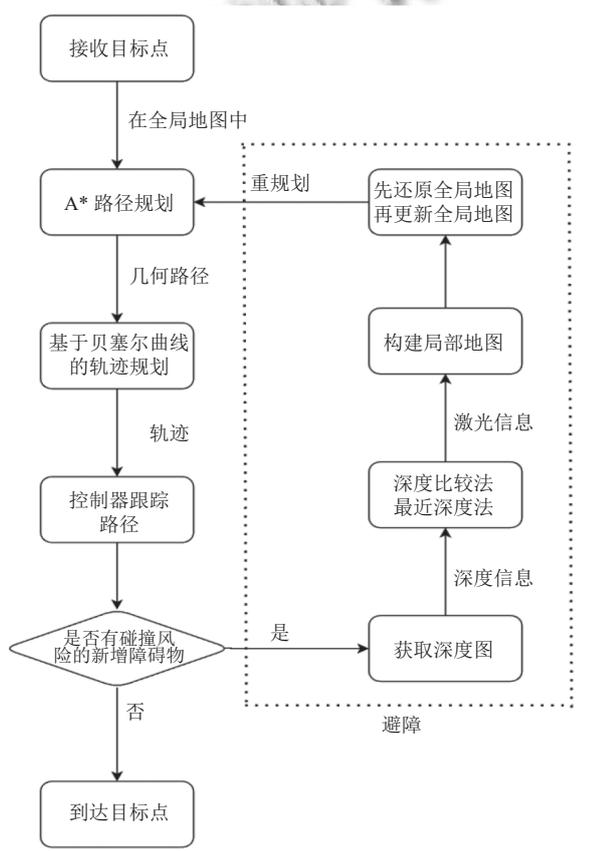


图 10 基于视觉的室内机器人导航避障方法

控制器跟踪轨迹的过程中, 会实时对 2 m 范围内的障碍物与轨迹之间进行碰撞风险检测, 如果有碰撞风险的障碍物距离机器人在 0.8–2 m 范围内, 机器人会继续行走; 如果障碍物出现在距离机器人 0.8 m 范围

内, 机器人会立即停止运动. 无论触发上述两种行为的哪一种, 机器人都会通过相机持续观测 2 s, 结合二维概率栅格地图构建原理, 如果障碍物对应的一系列栅格的置信度小于空闲阈值, 说明该障碍物是动态障碍物, 当发现障碍物碰撞检测通过时, 说明动态障碍物已经离开, 不会碰撞, 机器人可以继续跟踪轨迹前进; 如果障碍物对应的栅格的置信度大于占据阈值, 说明此处是静态障碍物, 则需要后续避障, 机器人会将当前深度图转换为激光信息, 构建局部地图, 结合机器人的当前的位姿信息, 将局部地图更新到全局地图上, 最后在更新后的全局地图的基础上重新进行 A* 路径规划和基于贝塞尔曲线的轨迹规划, 得到避障轨迹, 控制器控制机器人沿避障轨迹行走, 完成避障.

4 实验结果与分析

为了验证该导航避障方案的可行性, 在仿真与实际室内环境下进行了实验验证. 首先, 在稀疏和密集障碍物环境中验证了基于 Bezier 曲线的轨迹规划效果, 并在两种环境中对安全走廊改进前后的轨迹效果进行对比分析, 尤其是轨迹偏移量; 然后对改进安全走廊的膨胀半径 R_{range} 与轨迹偏移量之间的关系进行实验分析. 其次, 在机器人平台上验证了本文视觉数据处理方法能够减少对计算资源的需求, 并证实导航与避障方法的有效性, 机器人能够有效识别低矮和悬空、动态和静态障碍物并完成避障.

首先在 Gazebo 仿真环境下建立密集障碍物环境和稀疏障碍物环境, 并建立导航所需全局地图. 图 11 中橙色圆圈为圆形底盘机器人位置, 在 Rviz 中发布目标点, 可以很好地生成轨迹; 蓝色路径是 A* 规划的几何路径; 红色路径为本文基于贝塞尔曲线轨迹规划优化出来的轨迹; 一系列蓝色矩形即为安全走廊. 可以发现通过安全走廊的约束, Bezier 曲线优化出的轨迹一定在走廊范围内, 能完全避免机器人与障碍物发生碰撞, 同时可以看出红色轨迹明显, 蓝色几何路径平滑度更高, 更适合机器人的跟踪控制行走.

从图 11 可以看出, 红色轨迹与蓝色几何路径距离偏移较大, 但通常希望规划出的轨迹平滑且与几何路径尽可能贴近, 于是本文进一步改进了安全走廊的生成, 膨胀半径 R_{range} 为 2 个栅格, 改进后的效果如图 12. 可以发现在 A* 路径相同的情况下, 改进安全走廊的生成后, 优化后的轨迹明显更贴近 A* 路径, 同时轨迹满足平滑性要求.

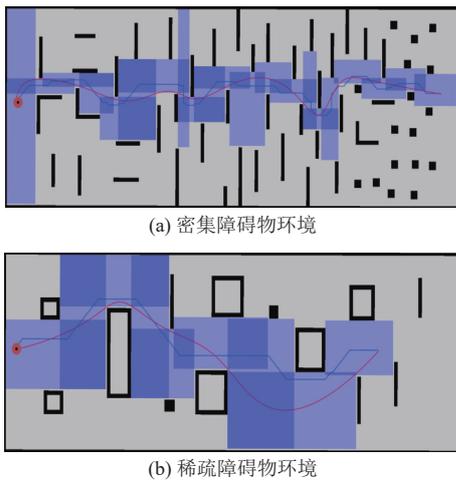


图 11 A*路径与贝塞尔轨迹规划图(原始安全走廊)

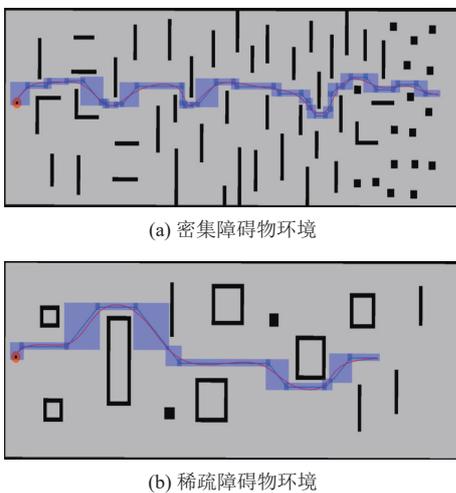


图 12 A*路径与贝塞尔轨迹规划图(改进安全走廊)

在密集和稀疏障碍物环境下,将安全走廊改进前后生成的轨迹在长度、平滑性和规划耗时这三个方面进行对比,如表1、表2所示.轨迹的平滑性可以用相邻路径点之间的角度变化来衡量^[20],假设希望轨迹尽可能接近直线,则相邻路径点构成的向量夹角尽可能接近 0° ,向量夹角的余弦值接近1,本文用轨迹相邻路径点构成的向量夹角的余弦值与1的差值来衡量平滑性,平滑性越好,差值越小,反之差值越大.最后将安全走廊改进前后的轨迹偏移量进行对比,如图13、表3和表4所示.图13中红色曲线为原始安全走廊下生成轨迹的偏移量曲线;蓝色曲线为改进安全走廊下生成轨迹的偏移量曲线.由表1和表3可知,在密集障碍物环境下,安全走廊改进后的轨迹长度、平滑性、规划耗时均有所提高,但与改进前的十分接近.但在轨迹偏移量方面,改进安全走廊生成的轨迹最大偏移量减少

0.267 m,平均偏移量减少 0.099 m,偏移量标准差减少 0.062 m,平均轨迹偏移量减少 63.9%,效果明显提升.由表2和表4可以看出,在稀疏障碍物环境下,安全走廊改进后轨迹长度、平滑性、规划耗时也均有所提高且与改进前的十分接近.但在轨迹偏移量方面,改进安全走廊生成的轨迹最大偏移量减少 0.969 m,平均偏移量减少 0.235 m,偏移量标准差减少 0.262 m,平均轨迹偏移量减少 77.8%,效果提升显著,说明本文改进安全走廊生成的轨迹在原有轨迹效果的基础上大幅降低了轨迹偏移量,在稀疏障碍物环境中效果更为明显.

表 1 密集障碍物环境下轨迹长度、平滑性和规划耗时表

环境	长度 (m)	平滑性	耗时 (s)
原始安全走廊	20.41	0.211	0.181
改进安全走廊	20.63	0.252	0.196

表 2 稀疏障碍物环境下轨迹长度、平滑性和规划耗时表

环境	长度 (m)	平滑性	耗时 (s)
原始安全走廊	17.31	0.043	0.051
改进安全走廊	17.43	0.052	0.059

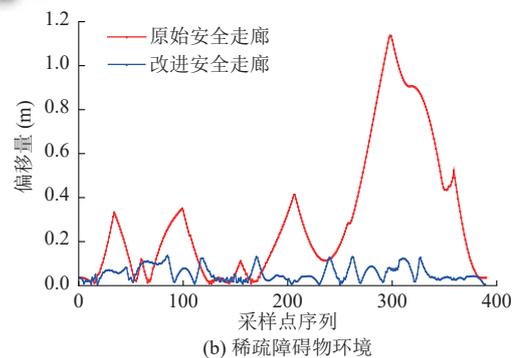
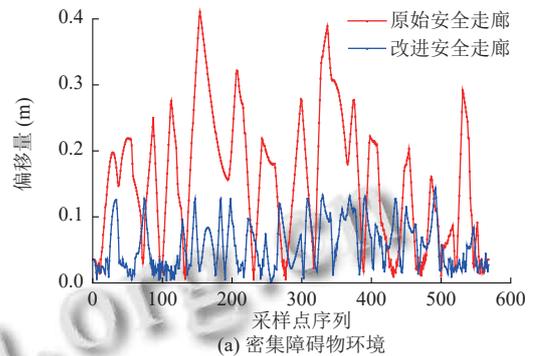


图 13 轨迹偏移量对比曲线

表 3 密集障碍物环境下轨迹偏移量对比表 (m)

环境	最大值	平均值	标准差
原始安全走廊	0.411	0.155	0.097
改进安全走廊	0.144	0.056	0.035

表4 稀疏障碍物环境下轨迹偏移量对比表 (m)

环境	最大值	平均值	标准差
原始安全走廊	1.148	0.302	0.301
改进安全走廊	0.179	0.067	0.039

为了衡量改进安全走廊的膨胀半径 R_{range} 的大小与轨迹偏移量之间的关系, 在稀疏障碍物环境中不同膨胀半径下测量所生成轨迹与 A*几何路径的偏移量. 表5描述了在稀疏障碍物环境中使用改进安全走廊生成方式在不同膨胀半径 R_{range} 下的轨迹偏移量. 可以发现随着膨胀半径的增加, 轨迹偏移量的最大值、平均值和标准差均逐渐增大, 成正比关系, 膨胀半径 R_{range} 的大小直接影响着轨迹偏移量. 膨胀半径 R_{range} 不宜设置过大, 若设置过大则无法起到减小轨迹偏移量的效果; 膨胀半径 R_{range} 也不宜设置过小, 越小意味着解空间越小, 生成的轨迹平滑效果可能会下降. 因此, 可以根据导航实际需求设置膨胀半径 R_{range} 的大小, 以达到期望的轨迹效果.

为了验证本文所提基于视觉的室内移动机器人导航避障方法的有效性, 在机器人平台上进行实验验证. 先在小区域场景中进行导航避障, 以便更好分析避障过程. 测试避障功能时首先需要在初始全局地图上发布目标点, 机器人规划一条如图14(b)所示的, 从S点到T点的轨迹, 绿色点为机器人位置, 红色点为目标点. 当机器人跟踪轨迹行走时, 在机器人前进方向上设置8cm高的低矮障碍物, 如图15(a)红框选中区域所示, 机器人跟踪路径时不断通过相机观测利用深度比较法识别障碍物, 此时相机观测的灰度图和深度图如图15所示, 以图中A、B、C这3个像素点为例, 应用深度比较法, A、B两点是低矮障碍物上的两点, C点是低矮障碍物与地面的接触处, 认为是地面, 这3点的深度值对比如表6所示. 可以看出A、B两点的实际深度值分别为1.967m和1.896m, 明显小于对应的地面深度标定值, 可以确定有障碍物; C点的实际深度值与对应的深度标定值基本一致, 可以认为该像素点处没有障碍物. 将深度图中的每一个像素点对应的深度值 $d_{i,j}$ 与深度标定图中的深度值 $D_{i,j}$ 作对比, 应用深度比较法, 最终识别环境中的障碍物.

到T点的轨迹, 绿色点为机器人位置, 红色点为目标点. 当机器人跟踪轨迹行走时, 在机器人前进方向上设置8cm高的低矮障碍物, 如图15(a)红框选中区域所示, 机器人跟踪路径时不断通过相机观测利用深度比较法识别障碍物, 此时相机观测的灰度图和深度图如图15所示, 以图中A、B、C这3个像素点为例, 应用深度比较法, A、B两点是低矮障碍物上的两点, C点是低矮障碍物与地面的接触处, 认为是地面, 这3点的深度值对比如表6所示. 可以看出A、B两点的实际深度值分别为1.967m和1.896m, 明显小于对应的地面深度标定值, 可以确定有障碍物; C点的实际深度值与对应的深度标定值基本一致, 可以认为该像素点处没有障碍物. 将深度图中的每一个像素点对应的深度值 $d_{i,j}$ 与深度标定图中的深度值 $D_{i,j}$ 作对比, 应用深度比较法, 最终识别环境中的障碍物.

表5 膨胀半径 R_{range} 与轨迹偏移量关系表 (m)

R_{range} (栅格)	最大值	平均值	标准差
1	0.151	0.049	0.032
2	0.179	0.067	0.039
3	0.209	0.092	0.051
4	0.260	0.114	0.064
5	0.319	0.126	0.082
6	0.368	0.139	0.089

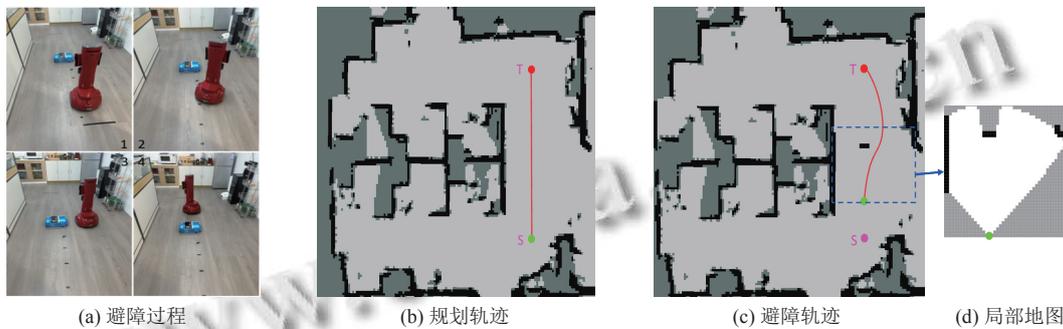


图14 基于视觉的机器人导航避障过程

表6 A、B、C的深度值对比 (m)

类型	A点	B点	C点
实际深度值	1.967	1.896	1.843
地面标定值	2.302	2.163	1.841

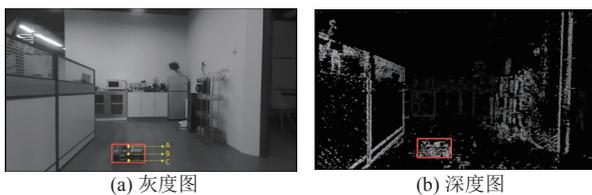


图15 低矮障碍物环境灰度图与深度图

另外对于障碍物的识别, 在RK3588S开发板上将本文深度比较法与点云法在CPU使用率、内存占用和处理时间等计算资源层面进行对比, 如表7所示. 与点云法相比, 本文方法在CPU使用率上降低了6.4%, 内存占用减少了30.51MB, 处理时间减少了0.151s. 可以看出减少了计算资源的需求, 尤其在处理时间上, 满足了对障碍物识别的实时性要求.

利用深度比较法识别到障碍物后, 将深度图像数据用最近深度法转换成激光数据, 并将激光数据中的

障碍物信息转化到栅格地图中,判断障碍物是否会与轨迹发生碰撞,发现图15(a)所示低矮障碍物有碰撞风险,但此时是第1次观测,对应栅格单元存在障碍物的置信度还未达到占据阈值;由于此时障碍物离机器人距离大于0.8 m,机器人会继续跟踪轨迹行走,相机持续观测,2 s后(如果2 s内机器人与障碍物距离小于0.8 m会停止运动,原地观测)发现有碰撞风险的障碍物对应栅格单元的置信度大于占据阈值,说明该障碍物是静态的,此时机器人停止运动,使用二维概率栅格地图的构建方法构建局部地图,如图14(d)所示.局部地图构建完成后,利用局部坐标系与全局坐标系的关系,将局部地图的白色区域和黑色区域更新维护到全局地图上,得到新的全局地图,如图14(c)所示,蓝色框选中区域为局部地图更新区域;绿色点为机器人当前定位点.接着在新的全局地图上重新轨迹规划,规划一条绕开临时障碍物的避障轨迹,如图14(c)中红色曲线所示,最终机器人跟踪新的避障轨迹成功到达目标点.机器人真实避障过程如图14(a)所示,机器人成功绕过低矮障碍物,并且与障碍物保持10 cm以上的距离,保证了导航的安全性.

表7 深度比较法与点云法计算资源对比表

方法	CPU (%)	内存 (MB)	耗时 (s)
点云法	72.3	1359.25	0.215
深度比较法	65.9	1328.74	0.064

为了进一步测试导航避障方法的有效性,在大区域室内场景中进一步测试静态悬空障碍物、静态低矮障碍物和动态障碍物的连续避障.首先在初始全局地图中轨迹规划一条从S点到T点的轨迹,记为轨迹1,如图16所示,然后在轨迹上A、B、C、D这4个点附近分别设置静态悬空障碍物、静态低矮障碍物、动态障碍物、先静态后动态障碍物.

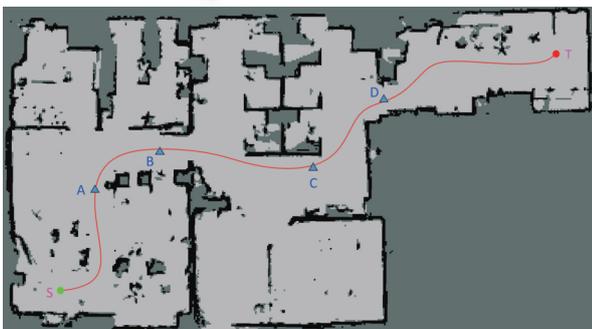


图16 轨迹1效果图

机器人从S点开始跟踪轨迹1行走,发现图16中A点附近存在有碰撞风险的静态临时悬空障碍物,如图17(a)所示,于是触发避障.使用深度比较法和最近深度法,可以发现使用深度图像数据转换成激光数据后,保留了三维环境数据,成功发现悬空障碍物,即图17(a)中蓝色的悬空延伸障碍物能够被构建在局部地图(对应局部地图的红色矩形框选中的黑色区域)中,此时局部地图与重规划的避障轨迹如图17(b)所示,该避障轨迹记为轨迹2.

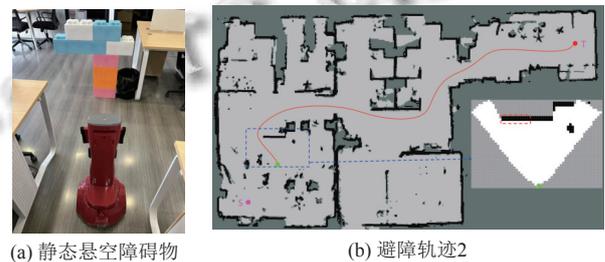


图17 静态悬空障碍物与避障轨迹2

机器人跟踪避障轨迹2行走,发现图16中B点附近存在有碰撞风险的静态临时低矮障碍物,如图18(a)所示,于是再次触发避障策略,此时局部地图与重规划的避障轨迹如图18(b)所示,记为避障轨迹3.

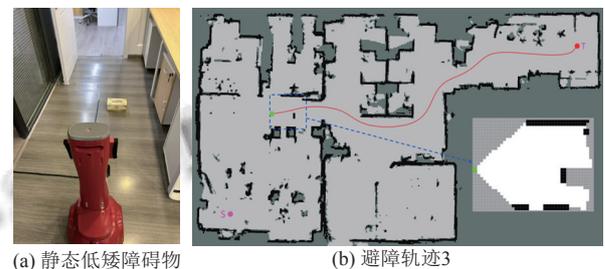


图18 静态低矮障碍物与避障轨迹3

机器人跟踪轨迹3行走,在图16中C点附近发现迎面走来的行人(动态障碍物).最开始行人与机器人距离大于0.8 m,如图19(a)所示,即便判断到行人是有碰撞风险的障碍物,但由于行人在走动,2 s内行人对应的栅格存在障碍物的置信度一直未大于占据阈值.确定行人是动态障碍物,机器人依旧沿轨迹3行走,只有当行人靠近机器人距离小于0.8 m且有碰撞风险时,即便置信度小于占据阈值,机器人也需制动停止,如图19(b)所示.此时再通过相机持续观测2 s,发现没有新的栅格置信度大于占据阈值,且不存在有碰撞风险的障碍物(行人离开),如图19(c)所示,机器人将继续

跟踪轨迹3行走,如图19(d)所示。

当机器人检测到图16中D点附近存在有碰撞风险的静态障碍物,如图20(a)所示,触发避障策略。由于障碍物已将门口堵住,重新规划轨迹时发现无法规划轨迹,如图20(b)所示,此时机器人会在30s内按1Hz频率不断构建局部地图并更新全局地图,尝试重新规划轨迹。在22s左右人为将D点的障碍物清除,机器人在1s内成功重新规划轨迹,记为轨迹4,如图20(d)所示,机器人跟踪轨迹4进入房间,如图20(c)所示,最终到达目标点T,很好地完成导航避障任务,整个导航避障过程机器人与障碍物均保持10cm以上的安全距离,保证了安全性。

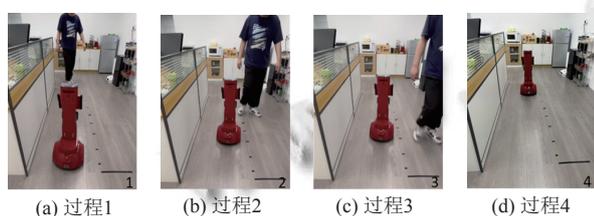


图19 动态障碍物避障过程

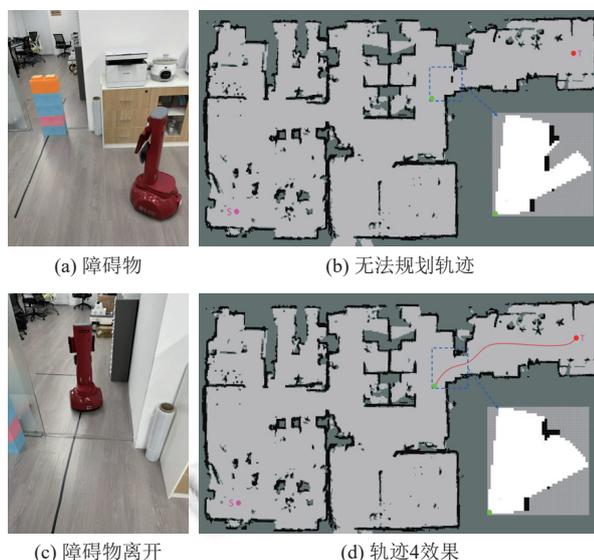


图20 先静态后动态障碍物避障过程

通过在真实机器人上的实验验证,证明了本文提出的“A*路径规划-贝塞尔轨迹规划-纯视觉避障”的导航避障方案的有效性,能够很好地满足室内移动机器人导航与避障的任务需求。

5 结论与展望

本文提出了一种“A*路径规划-贝塞尔轨迹规划-

纯视觉避障”的室内移动机器人导航避障方法,通过仿真及室内实验,证明了该方法可以很好地识别低矮障碍物,完成机器人导航所要求的规划、避障任务,在室内环境中较好的适应性。

(1) 根据A*算法规划获取到几何路径过后,基于贝塞尔曲线的轨迹规划方法有效解决了生成轨迹时可能会与障碍物发生碰撞的问题。同时改进安全走廊的生成,使轨迹更加贴近原始几何路径。在复杂障碍物环境下平均轨迹偏移量减少63.9%,稀疏障碍物环境下平均轨迹偏移量减少77.8%,解决了轨迹与A*几何路径偏移严重的问题。

(2) 最近深度法利用整张深度图将深度数据转为激光数据,深度比较法能够有效识别障碍物,排除地面避障过程的干扰,使计算资源需求整体下降;并根据得到的激光数据生成局部地图,将其更新到全局地图中,重新规划路径并生成避障轨迹,有效地实现了避障功能。在室内场景中进行的静态悬空障碍物、静态低矮障碍物和动态障碍物避障实验,可以很好地完成对障碍物的避障。

参考文献

- Patle BK, Babu LG, Pandey A, *et al.* A review: On path planning strategies for navigation of mobile robot. *Defence Technology*, 2019, 15(4): 582–606. [doi: 10.1016/j.dt.2019.04.011]
- Liu YS, Wu DF, You Z. An enhanced A* method incorporating an encrypted memory database for ASV efficient local path planning. *Mathematical Biosciences and Engineering*, 2024, 21(2): 2302–2322. [doi: 10.3934/mbe.2024101]
- Luo M, Hou XR, Yang J. Surface optimal path planning using an extended Dijkstra algorithm. *IEEE Access*, 2020, 8: 147827–147838. [doi: 10.1109/ACCESS.2020.3015976]
- 郭明阳, 刘爽. 基于模型预测控制的移动机器人户外导航方法. *华东理工大学学报(自然科学版)*, 2022, 48(2): 273–280.
- Hong ZH, Sun PF, Tong XH, *et al.* Improved A-star algorithm for long-distance off-road path planning using terrain data map. *ISPRS International Journal of Geo-information*, 2021, 10(11): 785. [doi: 10.3390/ijgi10110785]
- Iskander A, Elkassed O, El-badawy A. Minimum snap trajectory tracking for a quadrotor UAV using nonlinear model predictive control. *Proceedings of the 2nd Novel*

- Intelligent and Leading Emerging Sciences Conference (NILES). Giza: IEEE, 2020. 344–349.
- 7 沈跃, 李奕岑, 沈亚运, 等. 基于改进最小化 SNAP 的植保无人机作业轨迹优化算法. 农业工程学报, 2023, 39(17): 51–59. [doi: [10.11975/j.issn.1002-6819.202212003](https://doi.org/10.11975/j.issn.1002-6819.202212003)]
 - 8 Mellinger D, Kumar V. Minimum SNAP trajectory generation and control for quadrotors. Proceedings of the 2011 IEEE International Conference on Robotics and Automation. Shanghai: IEEE, 2011. 2520–2525.
 - 9 Liu SK, Watterson M, Mohta K, *et al.* Planning dynamically feasible trajectories for quadrotors using safe flight corridors in 3-D complex environments. IEEE Robotics and Automation Letters, 2017, 2(3): 1688–1695. [doi: [10.1109/LRA.2017.2663526](https://doi.org/10.1109/LRA.2017.2663526)]
 - 10 Lian L, Zong XJ, He K, *et al.* Trajectory optimization of unmanned surface vehicle based on improved minimum snap. Ocean Engineering, 2024, 302: 117719. [doi: [10.1016/j.oceaneng.2024.117719](https://doi.org/10.1016/j.oceaneng.2024.117719)]
 - 11 Zhang X, Lu YM, Ju ZY, *et al.* An integrated framework for motion planning and trajectory optimization of AGVs using spatio-temporal safety corridors. Control Engineering Practice, 2025, 159: 106297. [doi: [10.1016/j.conengprac.2025.106297](https://doi.org/10.1016/j.conengprac.2025.106297)]
 - 12 Labbé M, Michaud F. RTAB-map as an open-source lidar and visual simultaneous localization and mapping library for large-scale and long-term online operation. Journal of Field Robotics, 2019, 36(2): 416–446. [doi: [10.1002/rob.21831](https://doi.org/10.1002/rob.21831)]
 - 13 Mur-Artal R, Tardos JD. ORB-SLAM2: An open-source SLAM system for monocular, stereo, and RGB-D cameras. IEEE Transactions on Robotics, 2017, 33(5): 1255–1262. [doi: [10.1109/TRO.2017.2705103](https://doi.org/10.1109/TRO.2017.2705103)]
 - 14 Qin T, Li PL, Shen SJ. VINS-Mono: A robust and versatile monocular visual-inertial state estimator. IEEE Transactions on Robotics, 2018, 34(4): 1004–1020. [doi: [10.1109/TRO.2018.2853729](https://doi.org/10.1109/TRO.2018.2853729)]
 - 15 Xu Z, Liu X, Chen QL. Application of improved Astar algorithm in global path planning of unmanned vehicles. Proceedings of the 2019 Chinese Automation Congress (CAC). Hangzhou: IEEE, 2019. 2075–2080.
 - 16 Gao F, Wu W, Lin Y, *et al.* Online safe trajectory generation for quadrotors using fast marching method and Bernstein basis polynomial. Proceedings of the 2018 IEEE International Conference on Robotics and Automation (ICRA). Brisbane: IEEE, 2018. 344–351.
 - 17 Li HL, Luo YT, Wu J. Collision-free path planning for intelligent vehicles based on Bézier curve. IEEE Access, 2019, 7: 123334–123340. [doi: [10.1109/ACCESS.2019.2938179](https://doi.org/10.1109/ACCESS.2019.2938179)]
 - 18 徐靖贤. 自动驾驶车辆避障轨迹规划. 汽车实用技术, 2023, 48(4): 9–13.
 - 19 邱钟毓, 尹小恰, 陈凯, 等. 消防机器人城区内自主导航系统. 计算机系统应用, 2025, 34(1): 80–89. [doi: [10.15888/j.cnki.csa.009725](https://doi.org/10.15888/j.cnki.csa.009725)]
 - 20 Blažič S, Klančar G, Loknar MB, *et al.* Warehouse path planning using low-order Bézier curves with minimum-time optimization. IFAC-PapersOnLine, 2023, 56(2): 11815–11820. [doi: [10.1016/j.ifacol.2023.10.578](https://doi.org/10.1016/j.ifacol.2023.10.578)]

(校对责编: 王欣欣)