

基于改进 ORB_SLAM2 的机器人视觉导航方法

董蕊芳^{1,2} 王宇鹏^{1,2} 阚江明^{1,2}

(1. 北京林业大学工学院, 北京 100083; 2. 林业装备与自动化国家林业和草原局重点实验室, 北京 100083)

摘要: 提出了一种基于改进 ORB_SLAM2 的视觉导航方法。针对 ORB_SLAM2 构建的稀疏地图信息不充分、缺少占据信息、复用性差而无法直接用于导航的问题,对 ORB_SLAM2 算法进行了改进,融合环境的 3D、2D 占据特征以及路标点空间位置、视觉特征等多模态信息构建了包含定位、规划、交互图层的多图层地图以支撑机器人的精准定位和最优路径规划;针对机器人的自主导航任务,基于先验多图层地图建立约束进行机器人的位姿估计,融合运动约束进行机器人位姿优化,实现了基于先验地图的机器人精准定位,同时基于规划图层进行机器人的最优路径规划,提升了机器人的自主导航能力。在 TUM 数据集及北京鹫峰国家森林公园进行实验,结果表明:所构建的多图层地图提升了机器人定位的精度和效率,性能明显优于 RGB-D SLAM;基于先验地图的视觉定位方法能够进行机器人移动过程中精准、实时地定位,助力机器人按照所规划的路径实现准确的自主导航。

关键词: 机器人自主导航; 视觉 SLAM; 多图层地图; 先验地图; 路径规划

中图分类号: TP242.6

文献标识码: A

文章编号: 1000-1298(2022)10-0306-12

OSID:



Visual Navigation Method for Robot Based on Improved ORB_SLAM2

DONG Ruifang^{1,2} WANG Yupeng^{1,2} KAN Jiangming^{1,2}

(1. School of Technology, Beijing Forestry University, Beijing 100083, China

2. Key Laboratory of National Forestry and Grassland Administration for Forestry Equipment and Automation, Beijing 100083, China)

Abstract: Aimed at the problems of insufficient information and poor reusability of the sparse map constructed by ORB_SLAM2, a visual navigation method based on improved ORB_SLAM2 was proposed. It included two stages of building a multi-layer map and navigation. In the stage of building a multi-layer map, a local dense point cloud was calculated by the key frame of ORB_SLAM2, outliers were removed by radius filter and fast iterative closest point (FAST ICP) algorithm was used to register the processed point cloud. After that, 3D occupancy information was calculated by local dense point cloud; 3D occupancy information was extracted by means of the height of mobile robot in 3D space and 2D occupancy information was calculated by 2D mapping; 3D, 2D occupancy information and 3D, 2D features of the key frames were fused to generate a globally consistent multi-layer map. In navigation stage, according to the prior information of positioning layer, 2D features of the key frame were clustered to generate a visual dictionary, the visual dictionary was indexed according to the characteristics of current image to obtain the reference key frame; the initial pose was calculated by perspective-n-point (PnP) algorithm, and then reprojection error was used to construct inter-frame constraints, final result of localization was obtained through Gauss-Newton optimization; in planning layer, A* algorithm was used to plan path so that mobile robot visual navigation was realized. Verified by TUM dataset, the proposed method was about 50% faster than RGB-D SLAM, and the pose estimating was almost improved by 10%, the localization result based on prior map were consistent with the original map. In addition, the experiments on the real robot platform showed that the proposed method can construct a high-precision multi-layer map, and the error between the measured value of l_{AC} and the real value was 6.7%, and the error between the measured value of l_{BD} and the real value was 5.6%, and the fast and accurate navigation was achieved on the basis of multi-layer map.

Key words: autonomous robot navigation; visual SLAM; multi-layer map; prior map; path planning

收稿日期: 2022-04-15 修回日期: 2022-07-11

基金项目: 中央高校基本科研业务费专项资金项目(2021ZY72)和国家自然科学基金项目(32071680)

作者简介: 董蕊芳(1989—),女,讲师,主要从事视觉 SLAM 研究,E-mail: ruifang_dong@bjfu.edu.cn

通信作者: 阚江明(1976—),男,教授,主要从事机器人视觉信息处理研究,E-mail: kanjm@bjfu.edu.cn

0 引言

即时定位与地图构建 (Simultaneous localization and mapping, SLAM) 是机器人进行自主导航的先决条件,能实现在无全球导航卫星系统 (Global navigation satellite system, GNSS) 信号的情况下定位^[1]。视觉 SLAM 系统主要包括传感器信息获取模块、前端视觉里程计模块、后端优化模块和地图创建模块^[2-5]。视觉 SLAM 算法根据对图像处理方法的不同分为基于特征的方法和直接法^[6-8]。

基于特征的视觉 SLAM 指对输入的图像进行特征检测和提取,利用算法或者传感器获得特征点的深度,通过特征匹配构建帧与帧之间的共视关系,计算相机的位姿同时对 3D 特征进行优化建立地图。ORB_SLAM2^[9] 是第一个同时支持单目、双目、RGB-D 相机的视觉 SLAM 系统,该系统包含跟踪线程、局部地图构建线程以及回环检测线程,但其生成的稀疏点云信息不充分、复用性差而无法直接用于后续导航。文献[10]提出一种基于 ORB_SLAM2 实时构建网格地图的方法,根据稀疏点云直接生成 2D 占据网格地图。文献[11]提出一种用于机器人 RGB-D 视觉 SLAM 的算法,提取 ORB 特征,改进 RANSAC (Random sample consensus) 算法,与 GICP (Generalized-iterative closest point) 结合并加入随机闭环检测,构建场景环境的三维点云地图。文献[12]提出一种基于 ORB_SLAM2 的跳表地图构建算法,可实时构建 3D 网格地图。文献[13]对 ORB_SLAM2 进行拓展,开发地图保存功能。文献[14]提出一种针对机器人的导航地图制备方法,根据机器人在地面的运动约束通过点云地图可以推导出 2D 占据网格地图。

目前来看,已有众多研究针对机器人的自主导航,在 ORB_SLAM2 的基础上提升了建图性能或定位精度,但仍然没有一套完整的策略实现机器人自主导航。因此,本文提出一种基于改进 ORB_SLAM2 的视觉导航方法,以 ORB_SLAM2 的 RGB-D 模式为基础,融合环境的 3D、2D 占据特征以及路标点空间位置、视觉特征等多模态信息,构建包含定位、规划、交互图层的多图层地图以支撑机器人的精准定位和最优路径规划,并在多图层地图的基础上实现定位和导航。

本文提出的方法将实现多图层地图构建、基于先验地图的视觉定位、规划导航 3 个功能。以 ORB_SLAM2 关键帧为基础,构建多图层地图;以多图层地图作为先验信息,进行机器人实时定位;设定目标点后进行路径规划,最终实现机器人自主导航。

1 多图层地图构建

对 ORB_SLAM2 的 RGB-D 模式进行改进,根据其关键帧机制构建多图层地图。首先通过关键帧深度图计算局部稠密点云,通过分裂局部点云计算环境 3D 占据信息并更新交互图层;之后,对环境 3D 占据信息进行二维映射得到 2D 占据信息并以此来更新规划图层;与此同时,对关键帧中环境的路标点以及视觉特征建立对应关系,更新定位图层。将交互图层、规划图层以及定位图层进行坐标对齐,构建成融合了环境的 3D、2D 占据信息以及路标点和视觉特征的多图层地图,整体框架如图 1 所示。

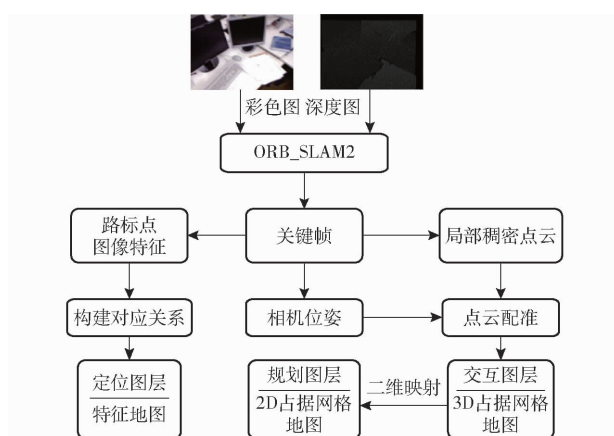


图 1 多图层地图框架

Fig. 1 Framework of multi-layer map

1.1 交互图层构建

交互图层负责实现导航过程中的人机交互,应具备丰富的外观信息,在此构建八叉树地图^[15]作为交互图层。构建过程包括局部稠密点云计算、点云配准以及环境 3D 占据信息更新 3 部分。

1.1.1 局部稠密点云计算

以 ORB_SLAM2 关键帧的深度图作为输入,通过 RGB-D 相机模型计算局部稠密点云,构建点云条件滤波器以剔除离群点。离群点剔除的方法如下:

取局部稠密点云其中一点 P , 取其最相近的 N 个点构成一个小样本, 设 $P(x, y, z)$, 在样本中另取一点 $P_i(x_i, y_i, z_i)$, 其中 $i = 0, 1, \dots, N$ 。计算 P 到 P_i 的距离 t_i 为

$$t_i = \sqrt{(x_i - x)^2 + (y_i - y)^2 + (z_i - z)^2} \quad (1)$$

平均距离 μ 计算公式为

$$\mu = \frac{1}{N} \sum_{i=1}^N t_i \quad (2)$$

假设局部稠密点云中共有 n 个小样本, 局部稠密点云整体的距离均值 $\bar{\mu}$ 和标准差 σ 计算公式为

$$\begin{cases} \bar{\mu} = \frac{1}{n} \sum_{i=1}^n \mu_i \\ \sigma = \sqrt{\frac{1}{n} \sum_{i=1}^n (\mu_i - \bar{\mu})^2} \end{cases} \quad (3)$$

若 μ_i 不在 $(\bar{\mu} - \sigma, \bar{\mu} + \sigma)$ 范围内,则认为该点是离群点,剔除该点。本文设定 $N = 50$,图 2(红色为离群点,绿色为保留点)所示为剔除离群点效果。

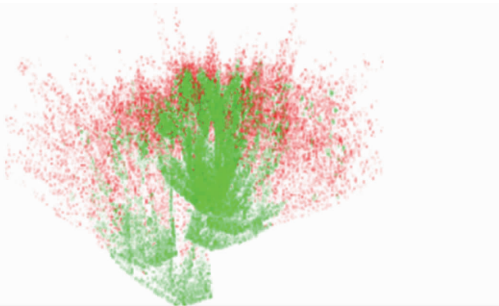


图 2 离群点剔除效果
Fig. 2 Outliers removed effect

以图 2 为例,局部稠密点云包含 32 876 个样本,记录每个小样本的均值 μ_i ,结果如图 3 所示。计算得出整体距离均值 $\bar{\mu} = 0.362\ 108\ \text{m}$,方差 $\sigma = 0.618\ 608\ \text{m}$,因此保留 μ_i 在 $0 \sim 0.980\ 716\ \text{m}$ 范围内的点云。

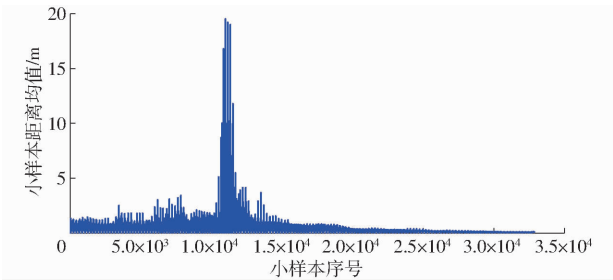


图 3 局部稠密点云的小样本距离均值
Fig. 3 Small sample distance mean of local dense point cloud

1.1.2 点云配准

采用 FAST ICP^[16](基于改进迭代最近点快速点云配准算法)进行配准,具体流程如下:假设图 2 中红色点云 A 为待配准点云,绿色点云 B 为目标点云,首先对 A 和 B 使用体素滤波器以降低分辨率,再对 A 和 B 分别用 ISS^[17](Intrinsic shape signatures)算法提取几何特征并计算内部形状描述子,用 RANSAC 算法迭代剔除外点。设 G 左上标为世界坐标系, C 左上标为相机坐标系,在观测点云 A 时刻的相机在世界坐标系下的位置为 ${}^G\boldsymbol{P}_A$ 、姿态为 ${}^G\boldsymbol{R}_A$,在观测点云 B 时刻的相机在世界坐标系下的位置为 ${}^G\boldsymbol{P}_B$ 、姿态为 ${}^G\boldsymbol{R}_B$,对 A 中的任意一点 ${}^C\boldsymbol{P}_{ai}$ 进行坐标变换,得到 ${}^G\boldsymbol{P}_{ai}$,即

$${}^G\boldsymbol{P}_{ai}^T = {}^G\boldsymbol{R}_A {}^C\boldsymbol{P}_{ai}^T + {}^G\boldsymbol{P}_A^T \quad (4)$$

之后,在 B 中寻找与 ${}^G\boldsymbol{P}_{ai}$ 距离最近的点 ${}^G\boldsymbol{P}_{bi}$,此

时认为 ${}^G\boldsymbol{P}_{ai}$ 和 ${}^G\boldsymbol{P}_{bi}$ 是一组匹配点。
根据上述方法,假设在 A 和 B 中有 n 对匹配点,定义 A 到 B 的变换矩阵为

$$\Delta\boldsymbol{T} = \begin{bmatrix} {}^G\boldsymbol{R}_A^T {}^G\boldsymbol{R}_B & \boldsymbol{R}_A^T ({}^G\boldsymbol{P}_B - {}^G\boldsymbol{P}_A) \\ 0 & 1 \end{bmatrix} \quad (5)$$

构造目标函数求解最优解

$$\Delta\boldsymbol{T}^* = \operatorname{argmin} \frac{1}{2} \|\boldsymbol{e}\|^2 = \frac{1}{2} \sum_{i=1}^n \|\boldsymbol{P}_{bi} - \boldsymbol{P}_{ai}\|_2^2 \quad (6)$$

式中 $\boldsymbol{e}$ ——匹配点位置误差
 $\boldsymbol{P}_{ai}$ ——A 中点云的坐标
 $\boldsymbol{P}_{bi}$ ——B 中与 A 对应点云的坐标

根据迭代次数以及迭代误差是否收敛(以前后两次迭代的变换矩阵的最大容差判断)等情况判断迭代是否结束,本文设定的迭代次数为 20 次,收敛范围为 0.000 1,配准结果如图 4(原始点云为红色,目标点云为绿色,配准后点云为蓝色)所示。

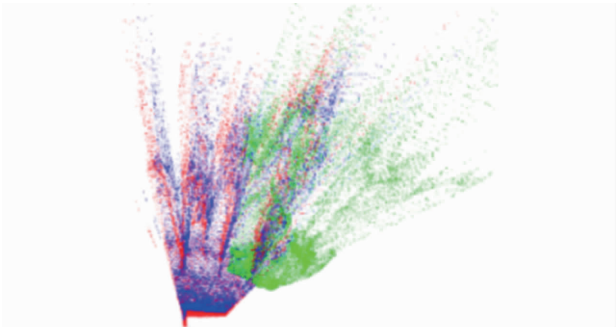


图 4 点云配准结果
Fig. 4 Result of point cloud registration

以图 4 为例,配准前的 $\Delta\boldsymbol{T}$ 为

$$\Delta\boldsymbol{T} = \begin{bmatrix} 1.00 & 0.02 & 0.05 & -0.07 \\ -0.02 & 1.00 & -0.01 & 0.11 \\ -0.05 & 0.01 & 1.00 & 0.47 \\ 0 & 0 & 0 & 1.00 \end{bmatrix} \quad (7)$$

配准后 $\Delta\boldsymbol{T}^*$ 为

$$\Delta\boldsymbol{T}^* = \begin{bmatrix} 1.00 & 0.02 & 0.08 & -0.08 \\ -0.19 & 1.00 & -0.07 & 0.13 \\ -0.08 & 0.07 & 1.00 & 0.60 \\ 0 & 0 & 0 & 1.00 \end{bmatrix} \quad (8)$$

1.1.3 环境 3D 占据信息更新

完成点云配准后,对点云构建包围盒并进行空间分裂。首先根据点云尺寸计算根节点包围盒尺寸,再根据设定分裂步长将根节点均匀划分为 8 等份,再依次判断是否到达递归深度、新的子节点中是否包含点云、子节点包含点云数量是否与父节点包含数量相同。点云空间分裂过程如图 5 所示。

根据点云空间分裂结果进行环境 3D 占据信息更新过程如下:假设 S 为空间分裂后的一个正方体,

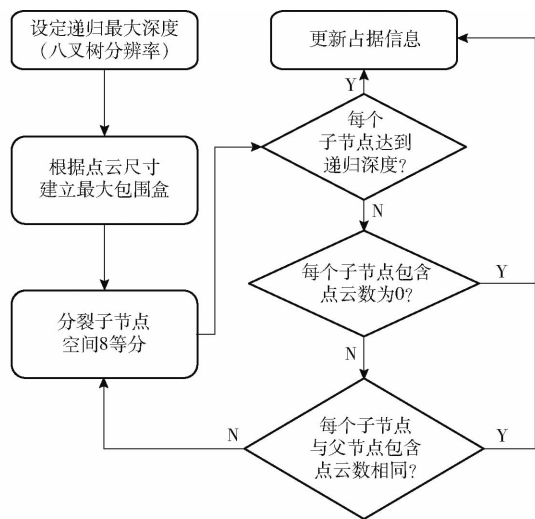


图 5 点云空间分裂流程

Fig. 5 Point cloud space segmentation process

其占据概率为 $p_s \in [0, 1]$, 依据其概率对数值 l_s 更新 S 的占据状态 p_s , S 的概率对数值 l_s 为

$$l_s = \ln \frac{p_s}{1 - p_s} \quad (9)$$

则有

$$p_s = \frac{e^{l_s}}{e^{l_s} + 1} \quad (10)$$

当观测到 S 的次数增加时, l_s 增加, 随着 l_s 增大, p_s 将趋近于 1, 表示 S 被占据的概率增大; 反之, 当观测到 S 的次数减少时, l_s 减小, 随着 l_s 减小 p_s 将趋近于 0, 表示 S 被占据的概率减小。

根据上述步骤, 可通过配准后的点云信息计算得到环境的 3D 占据信息, 之后将其融合至多图层地图中, 完成交互图层构建。

1.2 规划图层构建

规划图层用于机器人的路径规划, 而移动机器人的运动限制在 2D 空间, 因此本文以 2D 占据网格地图作为规划图层。在此对交互图层进行 2D 投影, 计算 2D 占据信息, 整体流程如下:

首先提取在交互图层中移动机器人运动空间部分的 3D 占据信息, 设机器人高为 h , 提取高度位于 $[0, h]$ 内的交互图层 3D 占据信息, 如图 6 所示。

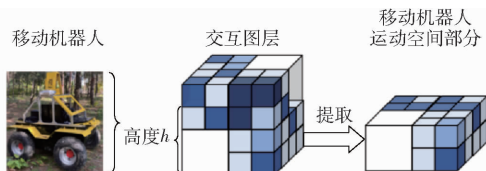


图 6 移动机器人运动空间占据信息提取

Fig. 6 Motion space occupancy information extraction of mobile robot

之后进行 2D 占据信息更新, 过程如图 7 所示。以机器人运动空间的占据信息作为集合 C 。定义以

$x-o-y$ 为平面, 在竖直方向的每一列正方体方格为集合 C 的子集 C_i , 记作

$$C = \{C_i, i = 1, 2, \dots, N\} \quad (11)$$

令左上标 $G2$ 表示 2D 空间, 左上标 $G3$ 表示 3D 空间, 则 C_i 表示为

$$C_i = \{ {}^{G3}P = (x, y, z), z = 0, 1, \dots, h \} \quad (12)$$

定义 C_i 中具有最大占据概率的点为 ${}^{G3}P_{p=\max}$, C_i 的 2D 映射点的坐标为 ${}^{G2}P = (x, y)$, 此时 ${}^{G2}P$ 的占据概率 $p({}^{G2}P)$ 计算更新为

$$p({}^{G2}P) = p({}^{G3}P_{p=\max}) \quad (13)$$

通过上述方法可对交互图层中的 3D 占据信息进行提取以及二维映射, 得到环境的 2D 占据信息并将其融合到多图层地图中, 最终完成规划图层的构建。

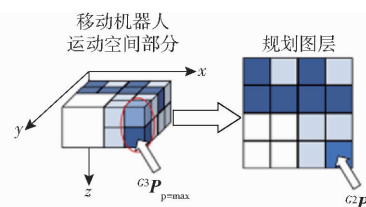


图 7 环境 2D 占据信息更新

Fig. 7 Environment 2D occupancy information update

1.3 定位图层构建

为了实现在已知地图中快速定位, 本文融合环境的路标点信息以及图像的特征信息构建定位图层, 使其能够为移动机器人快速定位提供基础。定位图层构建过程如图 8 所示。

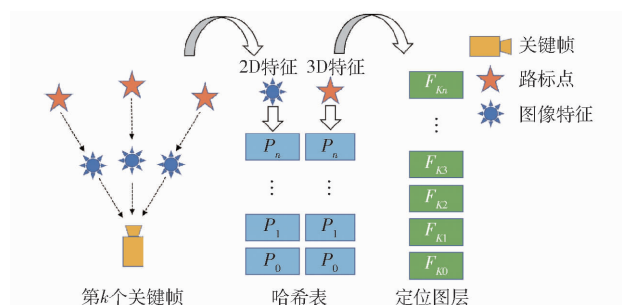


图 8 定位图层构建

Fig. 8 Localization layer construction

首先, 提取关键帧图像 2D 特征以及 3D 路标特征, 并将两者按照对应关系建立哈希表

$$H_{hashable}(F_K) = \{ [3D(P_i) - - 2D(P_i)] \} \quad (14)$$

式中, “- -” 表示一一对应。之后, 将每一个 $H_{hashable}(F_K)$ 存入一个向量列表中, 假设共有 n 个关键帧, 最终可得到定位图层

$$L = \{ H_{hashable}(F_K)_K, K = 0, 1, \dots, n \} \quad (15)$$

通过上述步骤, 利用关键帧的机制, 将环境的路标点信息和图像的特征信息按照对应关系进行融

合,最终实现定位图层的构建。

2 基于先验地图的视觉定位

本文提出的基于先验地图的机器人视觉定位方

法以定位图层为基础,整体框架如图 9 所示。定位方法包括 4 个主要部分:视觉字典构建、参考关键帧搜索、基于地图约束的机器人位姿估计、基于运动约束的机器人位姿优化。

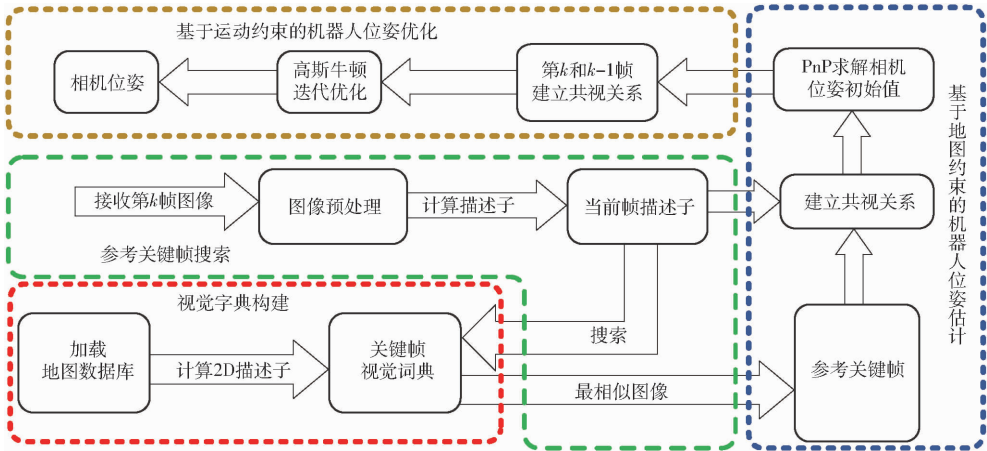


图 9 基于先验地图的视觉定位算法框架
Fig. 9 Framework of visual localization algorithm based on prior map

2.1 视觉字典构建

视觉字典构建类似于一个聚类问题,能够使机器人寻找数据中的规律。本文采用的方法是构建 k 叉树结构视觉字典,以保证对数级别的查找效率,下文中以 k 表示视觉字典的分支数, d 表示视觉字典的深度。

- (1)假设样本有 n 个特征点, k 个类簇中心。
- (2)根据样本中的特征计算得到 n 个描述子向量,计算样本中的每一个特征的描述向量到每个聚类中心的欧氏距离

$$D_i = \sqrt{(v_i - c_j)^2} \quad (i = 1, 2, \dots, n; j = 1, 2, \dots, k)$$

(16)

式中 v_i ——特征 i 的描述向量
 c_j ——第 j 个聚类中心向量

- 将 D_i 进行排序,取距离最短的作为它的归类。
- (3)计算所有类簇描述符的平均值作为新类簇中心。

(4)计算更新的类簇中心和之前的类簇中心变化值,如果每个类簇的变化值都很小,则迭代结束,否则循环步骤(2)、(3)。

视觉字典分支数 k 和视觉字典深度 d 这两个参数对视觉字典的构建耗时、大小以及查找效率起决定性作用,因此在应用过程中需根据实际情况去选择最优参数。

2.2 参考关键帧搜索

基于定位图层的参考关键帧搜索主要作用是快速索引出与当前图像具有最好共视关系的关键帧。首先,搜索参考关键帧需要计算当前图像的

BOW^[18-19] (Bag-of-words),检索 BOW 通常采用 TF-IDF^[20-21] (Term frequency - inverse document frequency)算法。此时假设在定位图层中有 N 个关键帧,每一个关键帧都可以用向量 V_N 表示,计算当前图像中的特征并以向量表示为 V_I ,计算当前图像与定位图层中每一个关键帧的相似度 s ,找到最相似的图像作为参考关键帧,公式为

$$s(V_I - V_N) = 2 \sum (|V_I| + |V_N| - |V_I - V_N|)$$

(17)

2.3 基于地图约束的机器人位姿估计

基于所检索到的参考关键帧,将当前帧和参考关键帧建立共视关系得到地图约束,进行机器人的位姿初始估计。首先,通过计算图像和参考关键帧描述向量的汉明距离进行匹配,匹配结果如图 10 所示。



图 10 图像描述子匹配
Fig. 10 Matches of image descriptors

根据图 10 可知,当环境特征相似度较高时,误匹配现象严重,因此进行误匹配点对的剔除。采用 RANSAC 剔除误匹配,实现方法如下:假设在图像 I1 中共有 m 个特征点,在图像 I2 中有 n 个特征点,根据描述子匹配计算得出匹配点有 k 对,在 k 对匹配点中随机抽取 4 个不共线的样本计算出单应性矩阵 H ,公式为

$$\lambda \begin{bmatrix} u'_i \\ v'_i \\ 1 \end{bmatrix} = \mathbf{H} \begin{bmatrix} u_i \\ v_i \\ 1 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} u_i \\ v_i \\ 1 \end{bmatrix} \quad (18)$$

式中 λ ——尺度因子

(u_i, v_i) ——I1 中第 i 个匹配的特征点

(u'_i, v'_i) ——I2 中第 i 个匹配的特征点

定义代价函数为

$$c_M = \sum_{i=0}^k \left[\left(u'_i - \frac{h_{11}u_i - h_{12}v_i + h_{13}}{h_{31}u_i - h_{32}v_i + h_{33}} \right)^2 + \left(v'_i - \frac{h_{21}u_i - h_{22}v_i + h_{23}}{h_{31}u_i - h_{32}v_i + h_{33}} \right)^2 \right] \quad (19)$$

重新随机抽取 4 个不共线的样本计算矩阵 $\mathbf{H}$, 迭代计算代价函数 c_M 直至到达全局最小, 此时得到单应矩阵 $\mathbf{H}$ 的最优解, 根据 $\mathbf{H}$ 进行投影剔除误匹配。剔除误匹配后结果如图 11 所示。

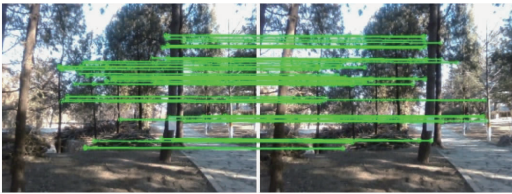


图 11 通过 RANSAC 算法剔除误匹配结果

Fig. 11 Wrong matches eliminated by RANSAC algorithm

建立当前图像特征和参考关键帧的 2D 特征对应关系后, 根据定位图层中 $\mathbf{H}_{\text{stable}}(F_{K_n})$ 索引对应 3D 点, 可得当前帧的 2D 特征与定位图层中的 3D 特征匹配关系, 如图 12(图中 R 表示相机的平移量, t 表示时间) 所示。用 PnP^[22] (Perspective-n-point) 求解当前帧位姿, 即可根据地图约束估计机器人位姿。

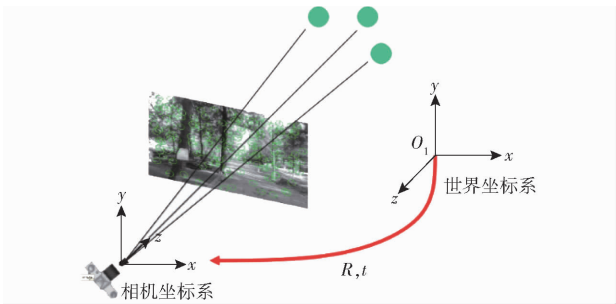


图 12 PnP 算法求解相机在先验地图中位姿

Fig. 12 Camera poses estimated by PnP algorithm

2.4 基于运动约束的机器人位姿优化

基于地图约束的机器人位姿估计结果存在一定误差, 说明相机的运动约束不可忽略, 因此本文在上述位姿估计结果的基础上, 基于运动约束进行机器人的位姿优化, 以提升定位精度。

假设在 k 帧, 已利用 PnP 求解出当前相机的初始位姿 x_k , 在此联系第 $k-1$ 帧建立相机的运动约束。首先寻找第 k 帧和第 $k-1$ 帧的共视 3D 地图特

征点, 设为集合 $\mathbf{P}_{k,k-1}$, 然后计算 $\mathbf{P}_{k,k-1}$ 在第 k 帧的重投影误差建立运动约束, 其误差目标函数

$$e(x_k, x_{k-1}, \mathbf{P}_{k,k-1}) = \sum \| \mathbf{h}(x_k, \mathbf{P}_{k,k-1}) - \mathbf{z}_k \|^2 \quad (20)$$

式中 e —— $\mathbf{P}_{k,k-1}$ 的重投影误差

$\mathbf{h}$ ——3D 特征点 $\mathbf{P}_{k,k-1}$ 到 x_k 的投影函数

$\mathbf{z}_k$ —— $\mathbf{P}_{k,k-1}$ 在第 k 帧的观测量, 即像素坐标

通过高斯牛顿法建立误差目标函数的正规方程

$$\mathbf{J}^T \mathbf{J} \Delta \mathbf{x} = -\mathbf{J}^T e(x_k, x_{k-1}, \mathbf{P}_{k,k-1}) \quad (21)$$

式中 $\Delta \mathbf{x}$ ——迭代步长

$\mathbf{J}$ ——误差雅可比矩阵

以 PnP 估计的相机位姿作为初始值进行迭代, 对于第 N 次迭代, 计算系统当前雅可比矩阵为 $\mathbf{J}_N$ 和误差 $e(x_k, x_{k-1}, \mathbf{P}_{k,k-1})_N$, 求解正规方程式 (21), 若在第 N 次时 $\Delta \mathbf{x}$ 足够小, 则停止, 否则继续迭代, 直到解出最优解。

3 基于多图层地图的机器人自主导航

本文提出的基于多图层地图的导航策略如图 13 所示, 具体包括:

(1) 机器人进行环境探索, 构建多图层地图。

(2) 基于先验地图进行初始定位, 计算机器人在先验地图中的初始位姿。

(3) 设定目标点, 基于规划图层利用 A* 算法规划最优路径。

(4) 根据实时定位结果, 采用 DWA (Dynamic window approach) 算法计算轨迹和运动策略^[23], 最终实现机器人自主导航。

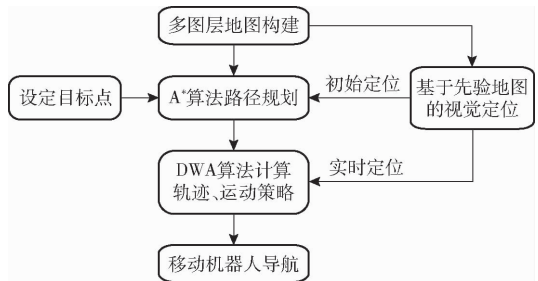


图 13 机器人自主导航框架

Fig. 13 Framework of autonomous navigation

3.1 基于规划图层的 A* 算法路径规划

机器人获取目标信息后, 需要根据建立的规划图层判断可行域, 再通过路径规划算法生成一条可通行全局安全的轨迹。A* 算法是最常用的基于图搜索的路径规划算法, 在二维栅格地图实现 A* 算法需首先构建两个二维向量组: *openlist* 和 *closelist*, 假设机器人当前定位点为 A , 目标点为 B , 根据图搜索的思想, 把栅格地图地中每一个方格 (分辨率

0.05 m)当作成一个节点,A* 算法搜索步骤如下:

(1)定义机器人水平移动代价值为 10,斜向移动的代价值为 14,代价函数 f 为

$$f(n) = g(n) + h(n) \tag{22}$$

式中 g ——起始点至 n 的代价

h —— n 至目标点的代价

以图 14 中点 C 为例, $g(C) = 24, h(C) = 44, f(C) = 68$ 。

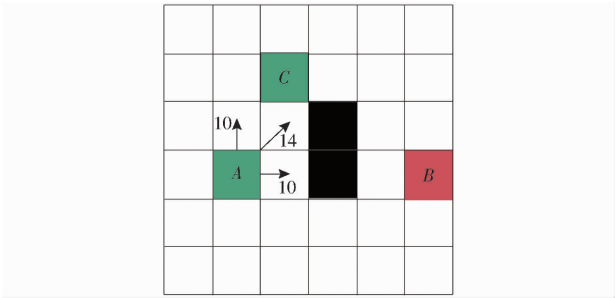


图 14 移动机器人在规划图层中移动的代价值
Fig. 14 Cost value of mobile robot moving in grid map

(2)将节点 A 放入 $closetlist$ 中,在节点 A 周围的方格中搜索可通行的区域,并计算每个方格的代价值 $f(n)$, A 的所有子节点放入 $openlist$ 中。

(3)在 $openlist$ 中取出 $f(n)$ 最小的节点并将其子节点加入 $openlist$,忽略障碍物和已在 $closetlist$ 中的节点。

(4)遍历上述步骤,如果节点 B 被加入 $openlist$ 中则表示搜索到最优路径,当所有栅格都被访问后节点 B 仍然没有被加入 $openlist$ 中,那么则表示没有路径可由节点 A 到达节点 B 。

(5)从目标点 B 回溯每个节点的父节点,最终得到最优路径。

上述 A* 算法进行路径搜索过程如图 15 所示。

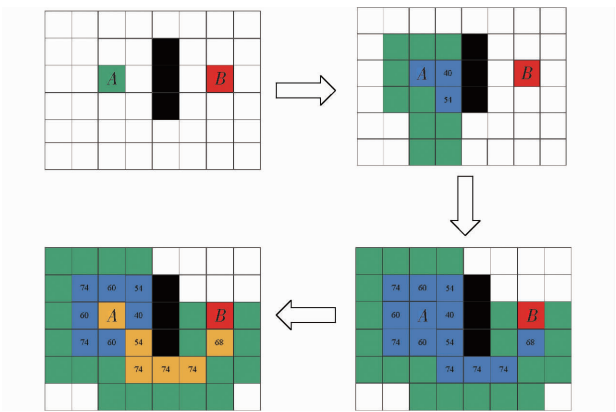


图 15 A* 算法示意图
Fig. 15 A* algorithm schematic

式(22)中的 $h(n)$ 也被称为 A* 算法的启发式函数,有多种表示方式,如:曼哈顿、对角线、欧几里得等。本文选用启发式对角线函数,其定义如下:

假设机器人对角线移动的代价值为 $h_d(n)$,定义

为

$$h_d(n) = \sqrt{2} \min(|n(x) - g(x)|, |n(y) - g(y)|) \tag{23}$$

式中 $n(x), n(y)$ ——可通行栅格 n 的 x, y 坐标
 $g(x), g(y)$ ——移动代价在 x 和 y 方向的分量

直线移动代价定义为

$$h_s(n) = \min(|n(x) - g(x)|, |n(y) - g(y)|) \tag{24}$$

此时的启发式函数为

$$h(n) = h_d(n) + h_s(n) - \sqrt{2}h_d(n) \tag{25}$$

在规划图层中设置两组相同的起点和终点,使用 A* 算法进行路径规划,结果见图 16。



图 16 A* 算法在规划图层路径规划
Fig. 16 A* algorithm in planning layer path planning

3.2 基于 DWA 算法的轨迹规划

机器人获得目标点后,根据 A* 算法规划出大致可行的路径,然后需要在设定好的路径计算运动策略。DWA 算法可根据规划的路径实时计算出速度。机器人应用 DWA 算法进行轨迹规划步骤如下:

(1)在速度空间 (v, ω) 中进行采样,假设当前机器人位于 (x, y) 处航向角为 θ ,根据不同的采样速度推算运动轨迹,则可推算出多条轨迹

$$\begin{cases} x' = x + v\Delta t \cos\theta \\ y' = y + v\Delta t \sin\theta \\ \theta' = \theta + \omega\Delta t \end{cases} \tag{26}$$

式中 (x', y') ——推演的位置
 θ' ——推演的航向角
 ω ——角速度 v ——线速度
 Δt ——采样时间间隔

(2)根据式(26)对每一条推算的轨迹进行评分 (G 表示得分)

$$G(v, \omega) = \lambda(\alpha \text{heading}(v, \omega) + \beta \text{dist}(v, \omega) + \gamma \text{vel}(v, \omega)) \tag{27}$$

其中, $\text{heading}(v, \omega)$ 表示机器人根据 DWA 算法推演出的最终航向角与目标点航向角的偏差, $\text{dist}(v, \omega)$ 表示机器人推演的最终位置和目标位置偏差,

$vel(v, \omega)$ 表示推演的停止速度与最终设定速度偏差。 α, β, γ 分别为航向角偏差权重、位置偏差权重以及速度偏差权重, λ 为整体权重, 保证机器人运动轨迹更佳平滑。为了使机器人能安全低速行驶, DWA 设定参数如表 1 所示。

表 1 DWA 算法参数设置

Tab.1 DWA algorithm parameters setting

机器人参数	最大值	最小值	分辨率
线速度/($\text{m}\cdot\text{s}^{-1}$)	0.5	-0.5	0.1
角速度/($\text{rad}\cdot\text{s}^{-1}$)	0.3	-0.3	0.15
最大角度偏差/rad	6.28	0	
最大距离偏差/m	0.5	0	
线速度样本数量	20	20	
角速度样本数量	20	20	

根据工作环境的不同, DWA 算法的朝向、位置以及速度权重选择的占比也不相同, 机器人在环境中工作需要能尽快安全到达指定的目标点, 因此最终位置的权重应该最高。设 $\lambda = 0.8, \alpha = 0.7, \beta = 0.2, \gamma = 0.1$, 在规划图层中测试 DWA 算法, 给定目标点, 根据 A* 算法生成一条全局轨迹(绿色轨迹), 再通过 DWA 算法进行航迹推演(黑色框为推演范围)得到最优的运动策略(红色轨迹), 如图 17 所示。

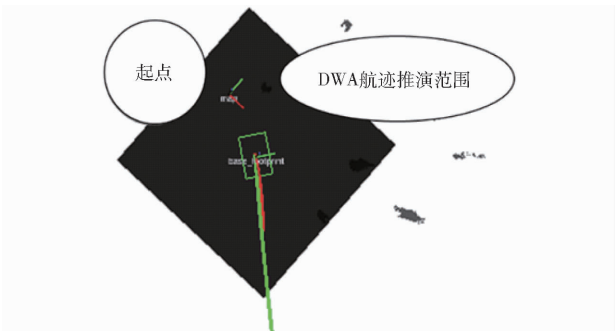


图 17 DWA 算法的生成轨迹

Fig. 17 Trajectory generated by DWA algorithm

4 实验

本文包括两组实验: 验证实验和实地实验。第 1 组选用 TUM 数据集^[24], 验证本文的多图层地图构建方法所估计轨迹的精度和实时性, 以及基于先验多图层地图的视觉定位方法估计结果与未知环境下 SLAM 定位结果的一致性。

第 2 组实验为北京鹫峰国家森林公园实地实验, 将本文的视觉导航方法应用于所搭建的轮式机器人。

4.1 TUM 数据集实验

TUM 数据集为 Kinect 视觉传感器采集的标准数据集, 该数据集中包含彩色图和深度图以及相机

运动过程中的 Ground truth(轨迹真值)。本组实验将通过 TUM 的 fr1/360、fr1/floor、fr1/rpy 序列验证。

4.1.1 多图层地图构建实验

本文对 ORB-SLAM2 算法进行改进, 构建多图层地图以支撑机器人的精准定位和最优路径规划。为验证其所构建多图层地图的优越性, 将其输出的位姿估计结果与 Ground truth 和经典的 RGB-D SLAM^[25] 结果进行比较, 结果如表 2 所示, 主要对比算法耗时和全局轨迹均方根误差两个指标。通过表 2 可知, 在 fr1/360 序列, 本文方法估计的轨迹精度比 RGB-D SLAM 提升 8.91%, 建图效率提高 49.38%; 在 fr1/floor 序列, 本文方法估计的轨迹精度比 RGB-D SLAM 提升 12.57%、建图效率提高 43.95%。总体而言, 本文方法的平均运行速度较 RGB-D SLAM 提升了 50%, 估计轨迹精度提升了 10%。因此本文提出的方法比 RGB-D SLAM 建立地图的效率更高, 轨迹更加准确。

表 2 TUM 数据集位姿估计结果

Tab.2 Results of pose estimating with TUM

序列	改进 ORB_SLAM2		RGB-D SLAM	
	算法耗时/s	均方根误差/m	算法耗时/s	均方根误差/m
fr1/360	157.41	1.84	310.96	2.02
fr1/floor	149.36	1.53	266.50	1.75
fr1/rpy	175.86	2.16	407.89	2.17

4.1.2 基于先验地图的视觉定位实验

由于 TUM 数据集仅提供了单次机器人运行数据, 因此, 本文进行基于先验地图的视觉定位实验, 思路为: 首先, 使用 TUM 数据集进行 SLAM, 构建多图层地图并输出机器人运行序列位姿; 然后, 在所构建地图的基础上, 再次输入相同的数据, 模拟机器人第 2 次进入相同的环境, 通过对比基于先验地图定位方法得到的序列位姿与 SLAM 过程估计的位姿之间偏差来验证算法的性能, 实验结果见表 3, 在 fr1/floor、fr1/rpy 这两个序列的定位结果与 SLAM 所估计轨迹的均方根误差仅有 0.01 m。这说明本文提出的基于先验多图层地图的视觉定位方法能够进行机器人的精确定位, 助力机器人实现准确的自主导航。

表 3 基于先验地图的视觉定位结果

Tab.3 Results of visual localization with prior map

序列	最大值	最小值	中值	标准差	均方根误差
fr1/360	0.38	0.02	0.01	0.04	0.05
fr1/floor	0.07	0.01	0.01	0.01	0.01
fr1/rpy	0.05	0.01	0.01	0.01	0.01

4.2 北京鹫峰国家森林公园实验

利用所搭建的轮式机器人在北京鹫峰国家森林公园进行实验。

机器人 为 四 轮 差 分 机 器 人,搭 载 的 视 觉 传 感 器 为 D435i 型 深 度 相 机。主 控 单 元 为 工 控 机,配 备 CPU 是 第 8 代 酷 睿 i5,256 GB 固 态 硬 盘,操 作 系 统 为 Ubuntu 18.04,如 图 18a 所 示。机 器 人 高 1.163 m,相 机 距 地 面 高 度 为 1.026 m,机 器 人 尺 寸 参 数 如 图 18b 所 示。

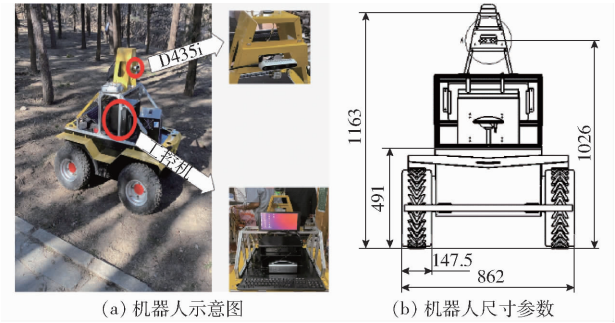


图 18 实验平台示意图

Fig. 18 Schematic of experimental platform

4.2.1 多图层地图构建实验

首先,需要根据机器人高度确定多图层地图中各个图层的坐标系位置。将交互图层按照机器人的高度分割成 3 部分:地面部分,地面到机器人高度范围部分,高于机器人部分。确定好提取范围后,对每一个关键帧生成的交互图层提取范围(−1.026 m, 0.110 m)内的包围盒,以 0.05 m 分辨率创建规划图层。在多图层地图中,定位图层和交互图层坐标轴一致,两者都以地图初始化时建立的初始坐标系为全局坐标系,而规划图层的坐标系在全局坐标系正下方 1.026 m 处。如图 19 所示, $O_1-x-y-z$ 为世界坐标系, $O_2-x-y-z$ 为规划图层坐标系。

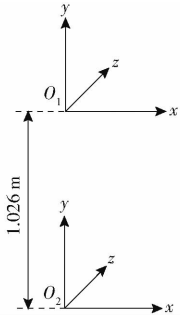


图 19 多图层地图坐标系示意图

Fig. 19 Schematic of coordinate system of multi-layer map

由于实验环境中难以获取相机轨迹真值,采用以下方式来检验建立地图的精度:实验场地为非规则图形,因此实验过程中设置了 4 个路标点,分别为 A、B、C、D,机器人在实验场地行走的路线为:由点 A 出发,按照 A→B→C→D→A 的路线行走,实时构建

多图层地图,图 20 为行走路线示意图。

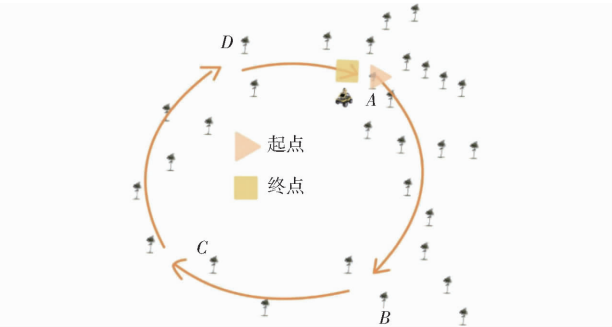


图 20 机器人行走路线

Fig. 20 Experimental design of robot walking route

构建的多层地图如图 21 所示,其中定位图层为特征点地图;交互图层为八叉树地图;规划图层为 2D 占据网格地图。

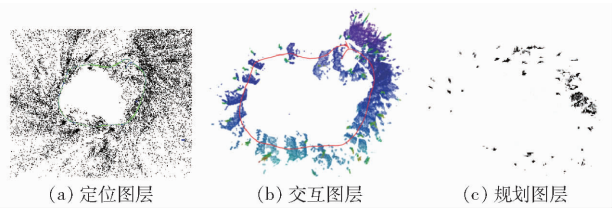


图 21 实验场景多图层地图

Fig. 21 Muti-layer map of experimental site

为方便统计,对交互图层中的树木进行编号,如图 22 所示,并分别用米尺测得实验场地中 AC、BD 的长度 l_{AC} 和 l_{BD} ,再通过多图层地图测量 AC、BD 之间的长度进行对比。

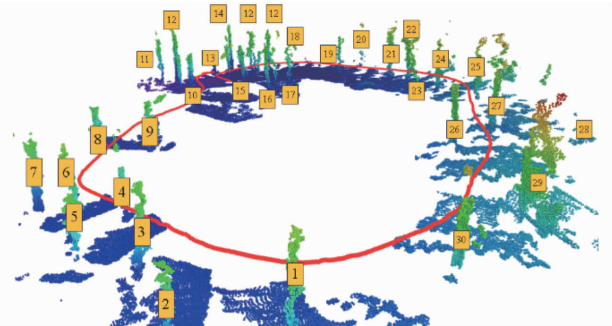


图 22 交互图层中树木编号

Fig. 22 Number trees in interactive layer

实验结果见表 4,结果表明, l_{AC} 的测量值与真实值之间的误差为 6.7%, l_{BD} 的测量值与真实值之间的误差为 5.6%,机器经过的路径周围共 30 棵树木,在地图中编号树木也为 30 棵。

表 4 实验结果

Tab. 4 Result of experimental site

参数	l_{AC}/m	l_{BD}/m	树木总数
真实值	21.33	19.40	30
测量值	19.90	18.30	30
误差	1.43	1.10	0

在真实环境中采用 RGB-D SLAM^[25] 算法实时构建地图时发现,该算法随着地图的增加会出现信息滞后的情况,而本文方法能够实时重建交互图层、规划图层以及定位图层,且地图精度和完整度都能得到保障,能为机器人自主导航提供准确的先验地图。

4.2.2 视觉字典构建实验

自主导航要求机器人快速准确地实现自身定位,在线构建视觉字典的时间以及字典大小直接影响机器人定位的效率以及精度,因此,本文对比选择 k (视觉字典分支数)、 d (视觉字典深度) 的最优参数来提升计算效率及精度。

实验中不同 k 、 d 构建字典的时间和生成字典的大小见表 5、6。

表 5 不同 k 、 d 构建视觉字典的时间

Tab.5 Cost time of k and d for generating visual dictionary s				
k	d			
	6	7	8	9
2	25.89	23.62	23.22	29.46
3	30.14	31.83	29.29	34.84
4	33.53	34.11	36.36	42.77

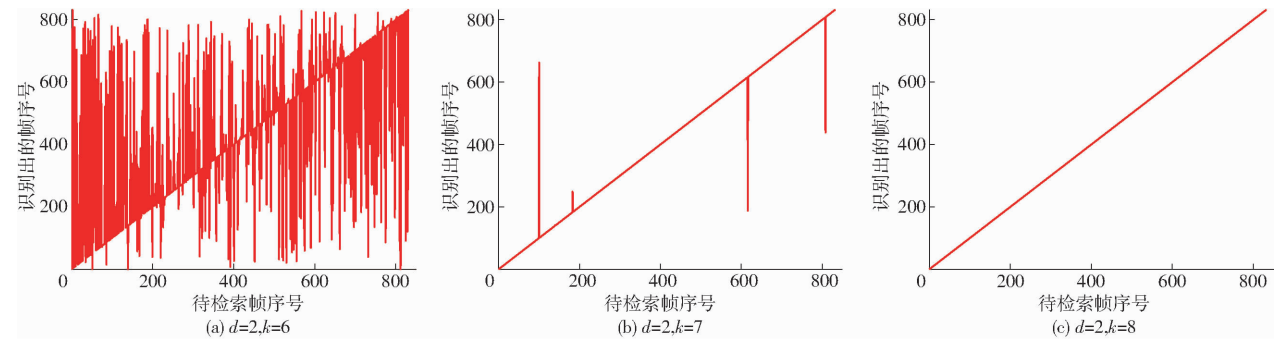


图 23 不同 k 、 d 的字典识别结果

Fig. 23 Recognition results using dictionaries with different k and d

具有最高的识别效率。因此,本文选择 $d=2$ 、 $k=9$ 作为最优参数进行计算。

表 7 k 、 d 对识别耗时的影响

Tab.7 Influence of k and d on recognition time ms				
k	d			
	6	7	8	9
2	9.4		11.2	12.3
3	16.8	18.8	19.4	18.7
4	18.3	16.9	15.6	13.4

4.2.3 基于先验地图的视觉定位实验

进行基于先验地图的视觉定位实验方式为:选择建图时的 4 个标记点 A 、 B 、 C 、 D (在地图中用红色点表示),使机器人分别行走 4 段路线: $A \rightarrow B$ 、 $B \rightarrow C$ 、 $C \rightarrow D$ 、 $D \rightarrow A$ (A 默认为坐标原点),即分别以 A 、 B 、 C 、 D 为起点进行实验,并分别记录下定位结果。

表 6 不同 k 、 d 构建视觉字典的大小

Tab.6 Memory size of k and d for generating visual dictionary kb				
k	d			
	6	7	8	9
2	2.0	2.7	3.5	4.5
3	13.1	20.3	29.7	41.8
4	76.7	137.5	229.7	361.8

对比不同 k 、 d 构建的 12 组视觉字典进行图像检索,验证识别效率。实验中对每一个关键帧进行顺序检索,按照关键帧出现的顺序进行验证,图 23 所示为检索结果,在使用 $d=2$ 、 $k=6$ 和 $d=2$ 、 $k=7$ 两种参数组合进行视觉字典构建时识别准确率分别为 51.62% 和 99.16%,在 $d=2$ 、 $k=8$ 时识别准确率达 100%。其余参数的视觉字典识别准确率均为 100%。

在验证参数 k 、 d 对计算效率的影响方面,分别记录利用以上 12 组视觉字典进行图像检索的时间,每组求和再计算平均耗时,结果见表 7。对比 k 、 d 对识别效率的影响,当 $d=2$ 、 $k=9$ 时,视觉字典的识别准确率为 100%,且在具有同等识别准确率当中

实验结果如图 24 所示。以图 24b 为例,机器人在实际地图中以 B 为起点,行走 $B \rightarrow C$ 段路线,但当机器人开始进入地图时,系统默认其初始位置为地图坐标原点(点 A),通过初始定位后,立刻确定自身位置在点 B ,并在后续运动过程中进行实时定位,最终结果表明机器人能够实现准确的定位。

4.2.4 基于多图层地图导航实验

本节实验按照第 3 节内容中所述步骤进行,机器人首先进行环境探索,构建实验环境的多图层地图,然后,根据定位图层和坐标变换确定机器人在规划图层的位置,即进行初始定位,如图 25a 所示;再次,根据任务目标点通过 A^* 算法生成最优路径,如图 25b 所示;基于规划的路径,采用 DWA 算法计算轨迹和运动策略,控制机器人移动,如图 25c 所示,同时,进行机器人的实时视觉定位;最终机器人到达

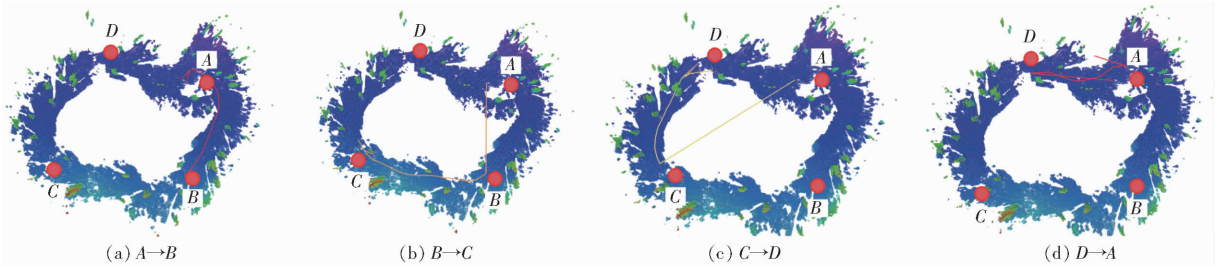


图 24 基于先验地图的视觉定位结果

Fig. 24 Result of visual localization based on prior maps

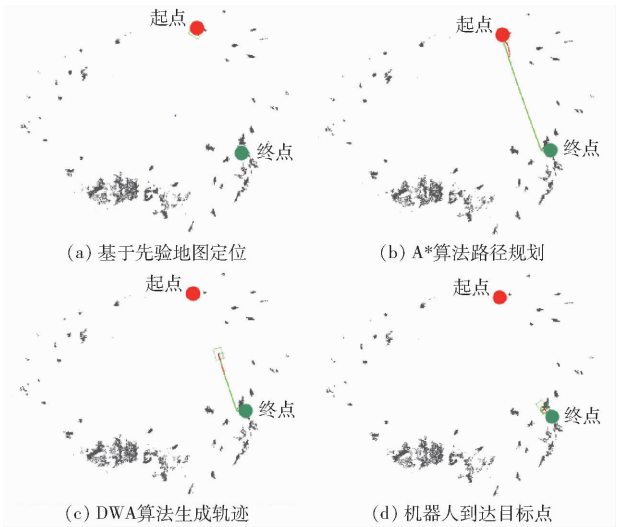


图 25 机器人自主导航过程

Fig. 25 Process of autonomous robot navigation

任务点,完成导航内容,如图 25d 所示。

为了更加直观地展示机器人在交互图层进行导航的过程,如图 26 所示,用红色点标记机器人导航的起点和终点,红色线段表示机器人的运动轨迹。

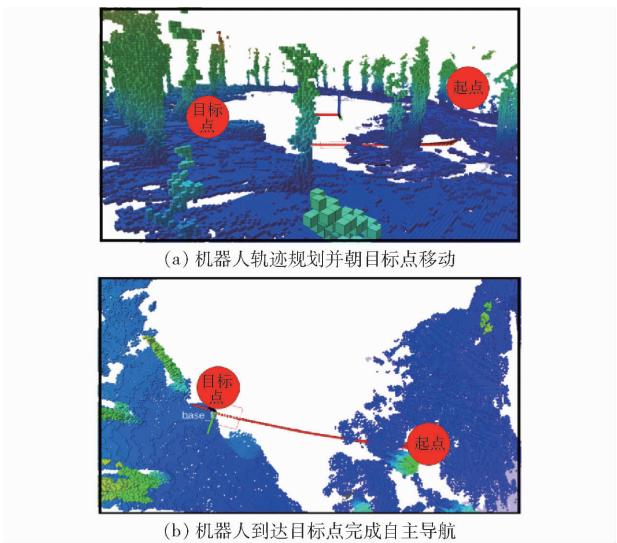


图 26 机器人在交互图层自主导航

Fig. 26 Autonomous robot navigation in interactive layer

交互图层能够显示较为完整的环境信息,设定目标点后,机器人根据规划图层进行路径规划,基于先验地图的视觉定位方法实时将机器人的位置进行反馈,实时检测机器人的位置,使整个导航的过程更加完整。

实验结果表明,本文提出的基于改进 ORB-SLAM2 的视觉导航方法能够构建完整的多图层地图,各图层分工明确,交互图层负责人机交互和生成规划图层,定位图层用来实现初始定位和导航过程中机器人的实时定位,规划图层用来生成最优路径;通过基于先验地图的视觉定位算法完成机器人的初始定位以及移动过程中的实时定位;最终完成机器人实时和精准地自主导航。

5 结论

(1)融合环境的 3D、2D 占据特征以及路标点空间位置、视觉特征等多模态信息构建包含定位、规划、交互等图层的地图,并通过实验证明本文方法在 fr1/360 序列比 RGB-D SLAM 轨迹精度提升 8.91%、建图效率提高 49.38%;在 fr1/floor 序列中表现比 RGB-D SLAM 精度提升 12.57%、效率提高 43.95%;最后在北京鹫峰国家森林公园构建多图层地图,对比分析得出 l_{AC} 的测量值与真实值误差为 6.7%, l_{BD} 的测量值与真实值误差为 5.6%。

(2)基于先验多图层地图建立约束进行机器人的位姿估计,融合运动约束进行机器人的位姿优化,实现了基于先验地图的机器人精确定位,利用 TUM 数据集验证得出,在 fr1/360 序列,定位轨迹与建图轨迹绝对轨迹误差相差 0.05 m,fr1/floor 序列相差 0.01 m,fr1/rpy 序列相差 0.01 m;在北京鹫峰国家森林公园,控制机器人走过 4 段路径,实验结果证明机器人在先验地图中能够利用视觉进行准确地定位。

(3)基于所构建的多图层地图进行机器人自主导航,实验结果证明机器人能够实时、准确地完成导航任务。

参 考 文 献

[1] 范永祥,冯仲科,陈盼盼,等. 基于 RGB-D SLAM 手机的森林样地调查系统研究[J]. 农业机械学报,2019,50(8):226-234.
FAN Yongxiang, FENG Zhongke, CHEN Panpan, et al. Research on forest plot survey system based on RGB-D SLAM mobile

- phone[J]. Transactions of the Chinese Society for Agricultural Machinery, 2019, 50(8):226–234. (in Chinese)
- [2] XIA L, CUI J, SHEN R, et al. A survey of image semantics-based visual simultaneous localization and mapping: application-oriented solutions to autonomous navigation of mobile robots[J]. International Journal of Advanced Robotic Systems, 2020, 17(3):41–58.
- [3] YAN Z, CHU S, DENG L. Visual SLAM based on instance segmentation in dynamic scenes[J]. Measurement Science and Technology, 2021, 32(9):95–113.
- [4] LUO Y, XIAO Y, ZHANG Y, et al. Detection of loop closure in visual SLAM: a stacked assorted auto-encoder based approach[J]. Optoelectronics Letters, 2021, 17(6):354–360.
- [5] ZHOU F, ZHANG L, DENG C, et al. Improved point-line feature based visual SLAM method for complex environments[J]. Sensors, 2021, 21(13):1–17.
- [6] 权美香, 朴松昊, 李国. 视觉 SLAM 综述[J]. 智能系统学报, 2016, 11(6):768–776.
QUAN Meixiang, PIAO Songhao, LI Guo. An overview of visual SLAM[J]. CAAI Transactions on Intelligent Systems, 2016, 11(6):768–776. (in Chinese)
- [7] 吴建清, 宋修广. 同步定位与建图技术发展综述[J]. 山东大学学报(工学版), 2021, 51(5):16–31.
WU Jianqing, SONG Xiuguang. Review on development of simultaneous localization and mapping technology[J]. Journal of Shandong University (Engineering Science), 2021, 51(5):16–31. (in Chinese)
- [8] 王霞, 左一凡. 视觉 SLAM 研究进展[J]. 智能系统学报, 2020, 15(5):825–834.
WANG Xia, ZUO Yifan. Advances in visual SLAM research[J]. CAAI Transactions on Intelligent Systems, 2020, 15(5):825–834. (in Chinese)
- [9] MUR-ARTAL R, TARDÓS J. ORB-SLAM2: an open-source SLAM system for monocular, stereo, and RGB-D cameras[J]. IEEE Transactions on Robotics, 2017, 33(5):1255–1262.
- [10] 陆建伟, 王耀力. 基于 ORB-SLAM2 的实时网格地图构建[J]. 计算机应用研究, 2019, 36(10):3124–3127.
LU Jianwei, WANG Yaoli. Real-time occupancy grid mapping using ORB-SLAM2[J]. Application Research of Computers, 2019, 36(10):3124–3127. (in Chinese)
- [11] 陈劭, 郭宇翔, 高天啸, 等. 移动机器人 RGB-D 视觉 SLAM 算法[J]. 农业机械学报, 2018, 49(10):38–45.
CHEN Shao, GUO Yuxiang, GAO Tianxiao, et al. RGB-D visual SLAM algorithm for mobile robots[J]. Transactions of the Chinese Society for Agricultural Machinery, 2018, 49(10):38–45. (in Chinese)
- [12] 王飞, 王耀力. 基于 ORB-SLAM2 的三维占据网格地图的实时构建[J]. 科学技术与工程, 2020, 20(1):239–245.
WANG Fei, WANG Yaoli. Real-time construction of 3D occupied grid map based on ORB-SLAM2[J]. Science Technology and Engineering, 2020, 20(1):239–245. (in Chinese)
- [13] NOBIS F, PAPANIKOLAOU O, BETZ J, et al. Persistent map saving for visual localization for autonomous vehicles: an ORB-SLAM2 extension[C]//15th IEEE International Conference on Ecological Vehicles and Renewable Energies (EVER), Monte-Carlo, 2020.
- [14] 杨蕴秀, 韩宝玲, 罗霄. 移动机器人同时定位与建图室内导航地图研究[J]. 科学技术与工程, 2021, 21(18):7597–7603.
YANG Yunxiu, HAN Baoling, LUO Xiao. Research on simultaneous localization and mapping of indoor navigation map for mobile robot[J]. Science Technology and Engineering, 2021, 21(18):7597–7603. (in Chinese)
- [15] HORNUNG A, WURM K M, BENNEWITZ M, et al. OctoMap: an efficient probabilistic 3D mapping framework based on octrees[J]. Autonomous Robots, 2013, 34(3):189–206.
- [16] XU G, PANG Y, BAI Z, et al. A fast point clouds registration algorithm for laser scanners[J]. Applied Sciences, 2021, 11(8):26–34.
- [17] ZHONG Y. Intrinsic shape signatures: a shape descriptor for 3D object recognition[C]//12th IEEE International Conference on Computer Vision Workshops (ICCV Workshops). Kyoto, 2009.
- [18] GAIVEZ-LOPEZ D, TARDOS J D. Bags of binary words for fast place recognition in image sequences[J]. IEEE Transactions on Robotics, 2012, 28(5):1188–1197.
- [19] NISTER D, STEWENIUS H. Scalable recognition with a vocabulary tree[C]//IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), New York, 2006.
- [20] SIVIC J, ZISSERMAN A. Video Google: a text retrieval approach to object matching in videos[C]//9th IEEE International Conference on Computer Vision. Nice, 2003.
- [21] ROBERTSON S. Understanding inverse document frequency: on theoretical arguments for IDF[J]. Journal of Documentation, 2004, 60(5):503–520.
- [22] LEPETIT V, MORENO-NOGUER F, FUA P. EPnP: an accurate $O(n)$ solution to the PnP problem[J]. International Journal of Computer Vision, 2009, 81(2):155–166.
- [23] 劳彩莲, 李鹏, 冯宇. 基于改进 A* 与 DWA 算法融合的温室机器人路径规划[J]. 农业机械学报, 2021, 52(1):14–22.
LAO Cailian, LI Peng, FENG Yu. Path planning of greenhouse robot based on fusion of improved A* algorithm and dynamic window approach[J]. Transactions of the Chinese Society for Agricultural Machinery, 2021, 52(1):14–22. (in Chinese)
- [24] SCHUBERT D, GOLL T, DEMMEL N, et al. The TUM VI benchmark for evaluating visual-inertial odometry[C]//IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). Madrid, 2018.
- [25] ENDRES F, HESS J, STURM J, et al. 3-D mapping with an RGB-D camera[J]. IEEE Transactions on Robotics, 2013, 30(1):177–187.