

# 基于 BoW 模型的增强 RGB-D SLAM 算法

荆树旭, 卢鹏宇, 翟晓惠, 高 涛

(长安大学 信息工程学院, 陕西 西安 710064)

**摘 要:** 即时定位与地图构建(simultaneous localization and mapping, SLAM)被认为是机器人自主运动的核心技术。针对目前的 RGB-D SLAM 算法实时性和鲁棒性差的问题,提出了一种增强的 RGB-D SLAM 算法。提取 RGB 图像的 ORB 特征描述子,然后利用 BoW(bag of word)模型缩小特征描述子的匹配范围从而提高算法的实时性;接着采用 PROSAC 算法结合 PnP 算法解算初始相机位姿并通过非线性优化的方式得到优化的相机位姿;利用 BoW 模型结合关键帧技术和结构一致性几何约束提高回环检测的鲁棒性;采用通用图优化工具 g2o 对位姿图进行优化,得到全局一致的位姿和点云;最后采用贪心三角化算法将点云转换成网格地图。针对 Fr1 数据集,该算法的平均定位误差为 0.079 7 m,每帧数据平均处理时间为 0.04 s。与 RGB-D SLAM 原始算法相比,该算法具有良好的实时性和鲁棒性,可以满足机器人实时 SLAM 的要求。

**关键词:** RGB-D SLAM; BoW 模型; PROSAC 算法; 回环检测; 位姿图优化; 网格地图

**中图分类号:** TP242.6

**文献标识码:** A

**文章编号:** 1673-629X(2021)04-0001-07

doi:10.3969/j.issn.1673-629X.2021.04.001

## Enhanced RGB-D SLAM Algorithm Based on BoW Model

JING Shu-xu, LU Peng-yu, ZHAI Xiao-hui, GAO Tao

(School of Information Engineering, Chang'an University, Xi'an 710064, China)

**Abstract:** Simultaneous localization and mapping (SLAM) is considered to be the core technology for autonomous motion of robots. An enhanced RGB-D SLAM algorithm is proposed to solve the problem of poor real-time performance and robustness of the current RGB-D SLAM algorithm. The ORB feature descriptor of RGB image is extracted, and then the BoW (bag of word) model is used to reduce the matching range of the feature descriptor to improve the real-time performance of the algorithm. Secondly, PROSAC algorithm and PnP algorithm are used to calculate the initial pose of the camera, and the nonlinear optimization is used to optimize the initial pose to obtain a high precision pose. The BoW model is combined with keyframe technique and structure-consistent geometric constraints to improve the robustness of loop detection. The general graph optimization tool g2o is used to optimize the pose-graph to get the globally consistent pose and point cloud. Finally, greedy-triangulation algorithm is used to transform point cloud into grid map. For the Fr1 data set, the average RMSE is 0.079 7 m, and each frame costs 0.04 s. Compared with the original algorithm of RGB-D SLAM, this algorithm has better real-time performance and robustness, which can meet the requirements of robot real-time SLAM.

**Key words:** RGB-D SLAM; BoW model; PROSAC algorithm; back-loop detection; pose-graph optimization; grid map

## 0 引 言

随着智能机器人技术的不断发展,研究者们希望机器人可以像人类一样,通过眼睛去观察和理解周围的世界,完成自身的定位和对周围环境的建模。这个问题被称为即时定位与地图构建(simultaneous localization and mapping, SLAM)<sup>[1-4]</sup>。视觉 SLAM<sup>[5]</sup>是指通过视觉传感器的方式完成即时定位与地图构建。相比使用激光雷达来获取周围环境信息,视觉传感器具有体积小,重量轻,功耗小,价格便宜等优点<sup>[4]</sup>。2010年,微软推出了可以同时获取彩色图和深度图的

相机 Kinect。由于其低廉的价格,使得 RGB-D 相机在视觉 SLAM 领域得到了广泛的应用<sup>[6-8]</sup>。

目前,国内外主流的方法是基于特征点法和图优化法的 RGB-D SLAM。2010年, Henry 等人<sup>[9]</sup>最早提出了一种基于深度相机的 SLAM 算法,该算法首先提取 SIFT 特征点,然后采用最近迭代点(iterative closest point, ICP)算法求解两帧图像之间的最优变换,最后利用图优化工具 TORO 进行全局优化。2012年, Henry 等人<sup>[10]</sup>提出了一种改进的算法,该算法利用重投影误差改进随机采样一致性(radom sample

收稿日期:2020-04-20

修回日期:2020-08-24

基金项目:国家自然科学基金(61703054);陕西省重点科技研发计划(2019-GY-039)

作者简介:荆树旭(1978-),男,副教授,研究方向为机器视觉和几何建模;卢鹏宇(1996-),男,硕士研究生,研究方向为计算机视觉与 SLAM。

consensus, RANSAC) 算法<sup>[11]</sup>的配准过程, 使用 Fast 特征提取和 Calonder 特征描述符代替之前的 SIFT, 用稀疏光速平差法 (sparse bundle adjustment, SBA) 代替 TORO 进行优化。2011 年, Newcombe 等人<sup>[12]</sup>提出的 KinectFusion 是最早能实现实时三维重建的方案, 可以在不需要 RGB 图只需要深度图的情况下就能实时地建立三维模型。2014 年, Whelan 等人<sup>[13]</sup>在 KinectFusion 的基础上增加了回环检测模块来增强鲁棒性。同年, Endres 等人<sup>[14]</sup>测评了 SIFT<sup>[15]</sup>、SURF<sup>[16]</sup>和 ORB<sup>[17]</sup>这 3 种方法在 RGB-D SLAM 中的表现, 在后端采用通用图优化工具 g2o 进行全局优化, 最后采用了八叉树地图 OctoMap 进行地图的构建。2017 年, Mur-Artal R 等人<sup>[18]</sup>提出了 ORB-SLAM2, 这是一个基于单目、双目和 RGB-D 相机的完整的 SLAM 系统。2018 年, 张震等人<sup>[19]</sup>利用 ORB 特征和词袋 (bag of words, BoW) 模型增强了 RGB-D SLAM 算法的鲁棒性。

上述算法存在的部分问题: 采用暴力匹配算法对特征点进行匹配, 需要每个特征与其他所有的特征进行匹配, 匹配范围大, 耗时严重。RANSAC 处理数据时具有随机性的特点, 忽略了待匹配点质量的差别。BoW 模型不考虑图像中视觉单词的位置信息, 会导致回环检测的精度不足。针对上述问题, 该文利用 BoW 模型结合型树层次结构, 将特征点划分为特征字典某一特定层的节点, 只有属于同一父节点的特征才是潜在的正确匹配, 从而缩小匹配范围, 提高算法的实时性。利用顺序抽样一致性 (progressive sample consensus, PROSAC) 算法<sup>[20]</sup>对图像匹配误差进行剔除并且求解相机位姿, 利用视觉词典选出一系列闭环候选帧, 对闭环候选帧进行严格的筛选, 从而筛选出真正的闭环帧, 提高算法的鲁棒性。

## 1 基于 BoW 模型的增强 RGB-D SLAM 算法

算法的整体流程如图 1 所示。在前端, 采取 ORB 算法提取图像的特征点, 基于 BoW 模型改进匹配, 在保证匹配效果的同时, 减少匹配的时间, 提高算法的实时性, 结合 PROSAC 算法和透视 n 点 (PnP) 算法计算相机的初始位姿变换, 利用 bundle adjustment (BA) 进行非线性优化, 得到更精确的相机位姿。通过关键帧筛选方案判断该帧是否成为关键帧。在后端, 通过 BoW 模型<sup>[21]</sup>对关键帧序列中的每一帧与当前关键帧图像进行相似度得分计算, 得分较高的列为闭环候选帧, 利用结构一致性对候选帧序列进行回环判定, 筛选出真正的闭环帧, 将闭环帧作为新的约束加入到位姿图中, 增强算法的鲁棒性。利用通用图优化工具 g2o

对位姿图进行全局优化, 得到全局一致的位姿。最后利用全局优化后的位姿和对应的点云数据, 采用贪心三角化算法重建出网格地图。

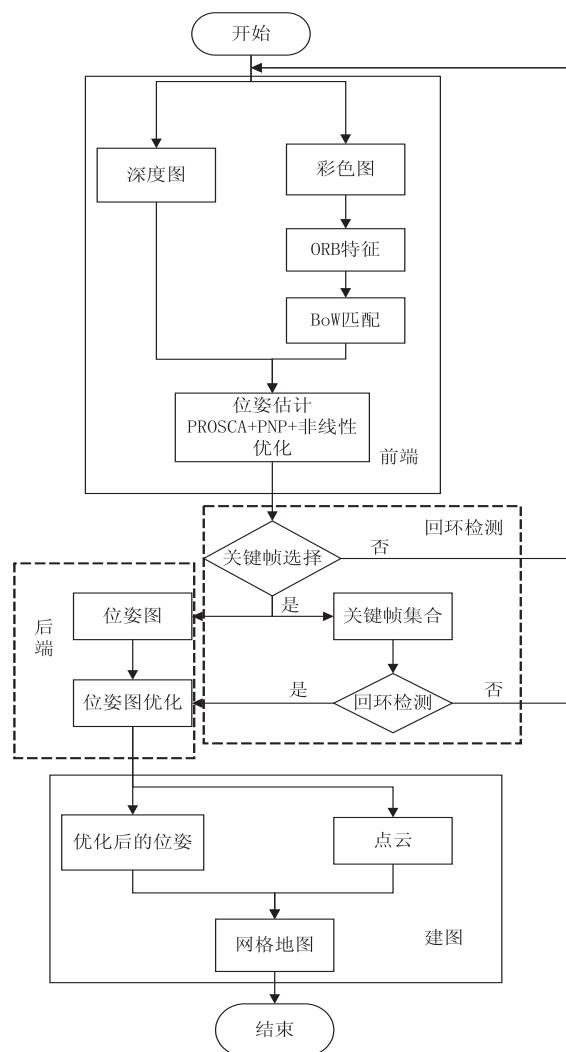


图 1 算法流程

### 1.1 ORB 特征提取与基于 BoW 的图像特征匹配

#### 1.1.1 ORB 特征提取

该文选用 ORB 特征算法进行特征提取。ORB 算法采用改进的 FAST 角点检测方法获取特征点, 采用 BRIEF 特征描述子生成特征描述符。与 SIFT 和 SURF 相比, ORB 算法在速度上拥有绝对的优势, 不需要使用 GPU 加速, 适合在嵌入式设备或缺少 GPU 运算能力的设备上使用。

#### 1.1.2 基于 BoW 的图像特征匹配

当两幅图像进行匹配时, 如果采用暴力匹配法, 将每幅内的每一个特征与另一幅图像内的每一个特征进行匹配, 其时间复杂度为  $O(N^2)$ 。而 BoW 的正向索引存储了图像特征描述子和它们在字典树中关联的节点, 这种方式可以极大地缩小图像特征的匹配范围, 加快匹配速度。算法步骤如下:

(1) 对任意两幅图像  $I_1$  和  $I_2$ , 提取它们的 ORB 特

征,得到它们的特征点和描述子,即  $(\text{keypoints1}_i, \text{descriptors1}_i)$  和  $(\text{keypoints2}_i, \text{descriptors2}_i)$ 。其中  $\text{keypoints1}_i$  表示  $I_1$  的特征点,  $\text{descriptors1}_i$  表示与  $\text{keypoints1}_i$  对应的特征描述子;  $\text{keypoints2}_i$  表示  $I_2$  的特征点,  $\text{descriptors2}_i$  表示与  $\text{keypoints2}_i$  对应的特征描述子。

(2) 对  $\text{descriptors1}_i$  和  $\text{descriptors2}_i$  求其对应的 BoW 特征向量  $(\text{node1}_i, \text{fl}_{ij})$  和  $(\text{node2}_i, \text{fl}_{ij})$ 。其中  $\text{node1}_i$  和  $\text{node2}_i$  表示第 1 层的节点,  $\text{fl}_{ij}$  表示以  $\text{node1}_i$  为父节点的特征,  $\text{fl}_{ij}$  表示以  $\text{node2}_i$  为父节点的特征。

(3) 当  $\text{node1}_i = \text{node2}_i$  时, 计算  $\text{fl}_{ij}$  和  $\text{fl}_{ij}$  对应的描述子  $\text{descripto1}_{ij}$  与  $\text{descripto2}_{ij}$  之间的汉明距离  $\text{Dist}$ , 设置最小值为  $\text{bestDist}_1$ , 次小值为  $\text{bestDist}_2$ 。

设定阈值  $H$  和  $h$ , 若满足:

$$\text{bestDist}_1 < H \quad (1)$$

$$\text{bestDist}_1 < h \times \text{bestDist}_2 \quad (2)$$

则认为该匹配是正确的匹配, 存入  $\text{Matches}$ 。

最终可得到鲁棒性强的匹配  $\text{Matches}$ ,  $\text{Matches}$  被用来进行下一步的图像配准和位姿估计。

## 1.2 位姿求解与优化

完成两幅图像的匹配后, 可以进行相机位姿的求解。该文使用 PROSAC 算法与 PnP 算法结合计算相机的初始位姿, 同时认为最邻近匹配的汉明距离越小, 匹配点的质量越高。得到相机的初始位姿后, 采用 BA 对初始位姿进行非线性优化, 得到精确的位姿, BA 问题即求解最小化重投影误差, 如式(3)所示。

$$(R, t)^* = \underset{(R, t)}{\operatorname{argmin}} \sum \rho(x_i - \Phi(RX_i + t))^2 \quad (3)$$

其中,  $(R, t)$  表示相机位姿,  $X_i$  表示空间点的坐标,  $x_i$  表示该点在像素内的坐标,  $\Phi$  表示相机内参,  $\rho$  表示 Huber 核函数。

BA 将在 g2o 中采用 Levenberg - Marquadt 实现。位姿求解与优化的具体算法如下:

(1) 对  $\text{Matches}$  中的匹配点按照  $\text{bestDist}_1$  的值从小到大进行排序。

(2) 在排序后的点中选取前  $n$  个点作为高质量匹配点作为 RANSAC 算法和 PnP 算法的输入, 求解出相机位姿的变换矩阵  $(R, t)$  和内点数  $\text{Inliers}$ 。

(3) 将  $(R, t)$  代入式(3)求得优化后的相机位姿  $(R, t)^*$ 。

## 1.3 关键帧选择

该文使用关键帧进行定位和建图, 通过计算帧间相对运动距离来判断该帧是否为关键帧。帧间相对运动距离的计算公式如下:

$$D = t + \min(2\pi - R, R) \quad (4)$$

其中,  $(R, t)$  表示相机位姿。

对当前帧  $\text{Frame}_{\text{curr}}$ , 判断该帧是否为关键帧的方法如下所示:

(1) 判断  $\text{Inliers}_i > \text{Inliers}_{\min}$ , 如果该式成立, 则进行下一步判断, 否则舍弃该帧。

(2) 将  $(R, t)^*$  作为初值输入式(4), 求得帧间相对运动距离  $D$ 。

(3) 设定阈值  $D_{\min}$  和  $D_{\max}$ , 若  $D_{\min} < D < D_{\max}$ , 则将帧加入关键帧序列。

该文引入了冗余关键帧删除机制, 有助于地图的构建和提高 SLAM 算法的鲁棒性。对当前关键帧  $\text{KeyFrame}_{\text{curr}}$ , 如果  $\text{KeyFrame}_{\text{curr}}$  上超过 90% 以上的特征点可以被至少其他 3 个关键帧检测到时, 认为  $\text{KeyFrame}_{\text{curr}}$  为冗余关键帧, 将其从关键帧序列里删除。

## 1.4 回环检测

该文通过 BoW 模型对关键帧序列进行回环检测, 首先计算待检索图像和数据库图像的 BoW 向量, 计算方式如下:

$$\begin{cases} q_i = n_i \times \omega_i \\ d_i = m_i \times \omega_i \end{cases} \quad (5)$$

其中,  $n_i$  和  $m_i$  分别表示待检索图像和数据库图像中单词  $i$  的数目,  $\omega_i$  表示单词  $i$  的权重。然后计算待检索图像和数据库图像之间的相似性分数, 计算方式如下:

$$s(q, d) = 1 - \frac{1}{2} \left( \left| \frac{q}{|q|} - \frac{d}{|d|} \right| \right) \quad (6)$$

利用式(6)可以计算当前帧  $\text{Frame}_{\text{curr}}$  与旧的关键帧  $\text{KeyFrame}_i$  之间的相似度。通过建立关键帧的倒排索引, 记录包含特定单词的关键帧。对于新的一帧图像, 首先计算该帧图像包含的单词表, 对于每一个单词, 通过倒排索引查找具有该单词的所有关键帧, 同时记录每个关键帧与当前帧的共同单词个数。对那些共同单词个数超过  $N$  的关键帧, 计算其与当前帧之间的相似性得分  $\text{Score}$ 。设定阈值  $\text{sTH}$ , 若相似度分数  $\text{Score}_i$  满足:

$$\text{Score}_i > \text{sTH} \times \text{Score}_{\max} \quad (7)$$

则将  $\langle \text{Frame}_{\text{curr}}, \text{KeyFrame}_i \rangle$  存储到候选闭环帧序列。

若闭环候选帧序列不为空, 对  $\langle \text{Frame}_{\text{curr}}, \text{KeyFrame}_i \rangle$  采用 1.1.2 节的方法得到匹配点对  $\text{Matches}$ , 之后利用 RANSAC 算法计算出基础矩阵  $F$  和内点数  $\text{inlier}$ , 利用  $F$  验证对极几何约束<sup>[22]</sup>, 如果满足几何约束关系并且  $\text{inlier}$  大于阈值  $\text{fTH}$ , 则认为  $\langle \text{Frame}_{\text{curr}}, \text{KeyFrame}_i \rangle$  构成回环。

## 1.5 全局优化与地图构建

该文通过位姿图进行全局优化, 将每个关键帧的相机位姿作为节点, 将两个位姿的变换关系  $T$  作为边构建位姿图。这样对于两个有边连接的节点, 会存在

一个误差:

$$e_{i,j} = x_i - T \times x_j \quad (8)$$

其中,  $x_i, x_j$  表示  $i, j$  节点的估计值。将所有的顶点和边考虑进来, 总的误差函数为:

$$C = \sum (e_{i,j}^T \Omega_{i,j} e_{i,j}) \quad (9)$$

其中,  $\Omega_{i,j}$  表示边的信息矩阵。这样整个问题被转换为一个最小二乘问题。该文采用 g2o 中的 Levenberg-Marquadt 法进行求解。

在相机运动过程中, 可能会出现运动丢失的情况。为此, 该文加入了重定位功能: 当连续  $M$  帧图像都出现失配的时候, 采用回环检测的方法寻找关键帧序列中与当前帧最为相似的图像, 之后通过 1.2 节的方法建立帧间约束并将当前帧设置为关键帧。

该文选用网格地图来进行稠密地图的构建, 其构建算法如下所示:

(1) 通过滤波对点云进行处理, 减少点云数据容量, 去除点云数据中的噪声、离群点。

(2) 通过重采样对物体表面进行平滑处理和漏洞修复。

(3) 计算点云法线, 并将点云位姿、颜色、法线信息合并到一起, 构建有向点云。

(4) 使用贪心投影三角化算法对有向点云进行三角化, 构建三角化网格地图。

## 2 实验结果与分析

为了验证算法的有效性, 该文利用 Sturm 等人<sup>[23]</sup>提供的 RGB-D 基准包进行实验, 该基准包中包含 Kinect 产生的彩色图像和深度图像序列, 以及机器人的真实运动位姿。同时, 文献[23]还提供了一个自动评估工具, 该工具可以自动计算出真实轨迹与算法估

计轨迹之间的均方根误差 (RMSE)。该文将从准确性和实时性与文献[19]、文献[24]和文献[25]进行对比。算法的实验平台为: Intel i5 9300H 处理器 (2.4 GHz), 16 GB RAM, 64 位 Ubuntu16.04 操作系统。

### 2.1 特征匹配结果分析

对 Fr1/xyz 数据包相邻的两帧图像进行特征匹配。表 1 为基于 BoW 模型的特征匹配算法和暴力匹配算法关于 Fr1/xyz 数据包的特征匹配时间对比, 图 2 所示为 BoW 匹配的结果。

表 1 Fr1/xyz 数据包特征匹配时间对比

匹配方法	时间/s
暴力匹配	10.7
BoW 匹配	6.2

由表 1 可知, 在 ORB 算法提取的特征点数量为 500, Fr1/xyz 数据包共产生 787 次匹配的情况下, 暴力匹配共使用了 10.7 s 的时间完成匹配, BoW 匹配共使用了 6.2 s 的时间完成匹配, 由此可见, BoW 匹配可以极大地减少特征匹配的时间。由图 2 可知, BoW 匹配的结果较好, 正确匹配数量多。

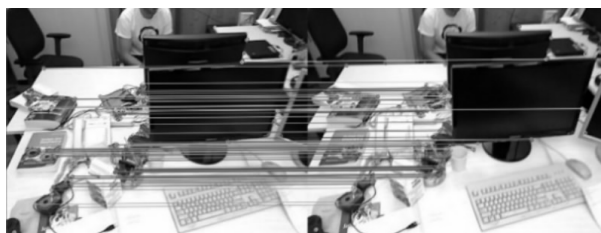


图 2 BoW 匹配结果

### 2.2 定位误差对比

该文选取了 Fr1 数据包中的 Fr1/xyz、Fr1/rpy、Fr1/desk 等 9 个数据集进行实验。表 2 为文中算法与文献[19]、文献[24]和文献[25]的均方根误差对比。

表 2 文中算法与文献[19]、文献[24]和文献[25]的均方根误差对比 m

数据包	长度	文献[19]	文献[24]	文献[25]	文中算法
Fr1/xyz	7.11	0.017	0.021	0.017	0.017
Fr1/rpy	1.66	0.029	0.042	0.032	0.024
Fr1/desk	9.26	0.039	0.049	0.045	0.029
Fr1/desk2	10.16	0.078	0.102	0.082	0.090
Fr1/floor	12.57	0.059	0.055	0.050	0.050
Fr1/room	15.99	0.209	0.219	0.250	0.139
Fr1/360	5.82		0.103	0.120	0.128
Fr1/plant	14.80		0.142	0.072	0.119
Fr1/teddy	15.71		0.138	0.090	0.075

由表 2 可以看出, 文中算法在 Fr1/xyz、Fr1/rpy、Fr1/desk 等 6 个数据集中的精度均优于文献[19]、文

献[24]和文献[25]中提出的算法。仅在 Fr1/desk2 数据集, 文中算法略低于文献[19]和文献[25], 但优于



文献[24];在 Fr1/360 数据集,文中算法略低于文献[24]和文献[25];在 Fr1/plant 数据集,文中算法略低于文献[25],但优于文献[24]。基于相同数据的集性能比较,文献[24]的平均 RMSE 为 0.096 8 m,文献[25]的平均 RMSE 为 0.084 2 m,文中算法的平均 RMSE 为 0.079 7 m,总体精度优于文献[24]和文献[25]。文献[19]在测试数据集的平均 RMSE 为 0.072 8 m,文中算法对应的数据集的平均 RMSE 为 0.066 5 m,总体精度优于文献[19]。根据以上统计结

果,验证了文中算法的总体精度优于文献[19]、文献[24]和文献[25],具有更好的准确性。

为了更加直观地显示文中算法的定位准确性,对估计轨迹与真实地面轨迹进行误差量化分析。图3是估计轨迹与真实轨迹的对比图,其中的(a)、(b)、(c)、(d)分别对应了表2的 Fr1/xyz、Fr1/desk、Fr1/floor、Fr1/room。表3为误差量化数据,包括最大值 max、最小值 min、中位数 median、均值 mean、标准差 std。

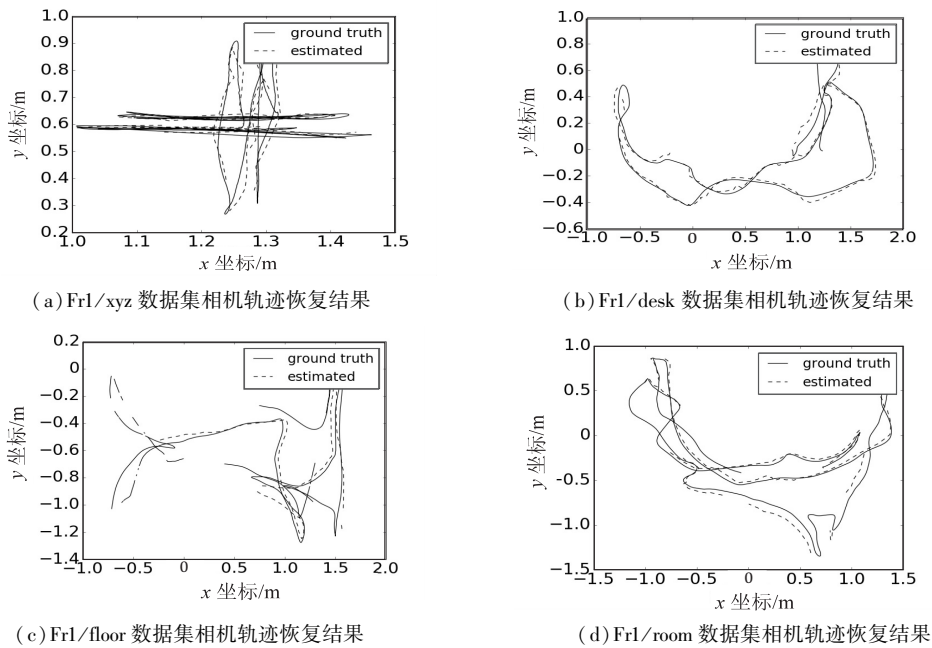


图3 真实轨迹与估计轨迹对比

表3 误差量化指标

数据包	max	min	median	mean	std
Fr1/xyz	0.036	0.001	0.012	0.015	0.008
Fr1/rpy	0.046	0.002	0.018	0.020	0.012
Fr1/desk	0.055	0.007	0.024	0.027	0.011
Fr1/desk2	0.137	0.022	0.058	0.081	0.055
Fr1/floor	0.089	0.014	0.044	0.048	0.016
Fr1/room	0.292	0.024	0.114	0.124	0.063
Fr1/360	0.294	0.024	0.094	0.111	0.062
Fr1/plant	0.253	0.034	0.098	0.106	0.054
Fr1/teddy	0.239	0.016	0.059	0.068	0.031

从图3可以看出,估计轨迹能较好地贴合真实轨迹,全局一致性较好。从表3可以看出在 Fr1 数据包中,误差值与变化波动都较小,总体误差在可接受范围。

### 2.3 运行速度对比

由于文献[25]只进行了特征点的提取,没有进行特征描述符与特征匹配,所以文中只与文献[19]和文

献[24]进行速度对比,对比结果如表4所示。

文献[24]处理一帧数据平均需要 0.35 s,即平均每秒钟处理约 3 帧数据,文献[19]处理一帧数据平均需要 0.11 s,即每秒钟处理约 10 帧数据,而文中算法处理一帧数据平均用时为 0.04 s,即每秒钟处理约 25 帧数据,是文献[24]的 8 倍,文献[19]的 2.5 倍。由于 Kinect 的帧率为 30 Hz,因此文中算法基本满足实时

性要求。

表 4 文中算法与文献[19]和文献[24]算法运行速度对比

s

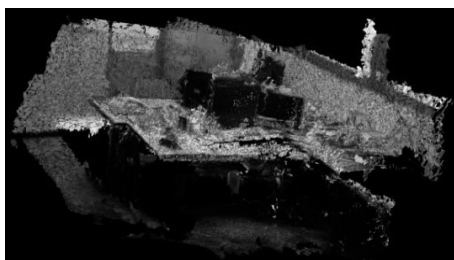
数据包	文献[19]	文献[24]	文中算法
Fr1/xyz	0.106	0.463	0.036
Fr1/rpy	0.113	0.354	0.048
Fr1/desk	0.122	0.346	0.050
Fr1/desk2	0.079	0.287	0.057
Fr1/floor	0.103	0.318	0.043
Fr1/room	0.092	0.402	0.042
Fr1/360		0.194	0.035
Fr1/plant		0.381	0.047
Fr1/teddy		0.398	0.041

## 2.4 地图构建评估

该文选用了 Fr1/xyz、Fr1/desk 和 Fr1/floor 数据包进行建图分析。图 4 的 (a)、(b) 和 (c) 分别表示使用 Fr1/xyz、Fr1/desk 和 Fr1/floor 数据包构建的三维网格地图。



(a) Fr1/xyz 数据集重建结果



(b) Fr1/desk 数据集重建结果



(c) Fr1/floor 数据集重建结果

图 4 场景重建效果图

Fr1/xyz 和 Fr1/desk 是办公室场景,为手持 Kinect 拍摄所得。由图 4(a) 和 (b) 可以看到,该场景内的办公桌、电脑、书籍和墙壁等都得到了很好的重建。图中的空洞部分是由于拍摄过程中没有扫描到那部分。

Fr1/floor 数据包由机器人携带 Kinect 拍摄所得。由图 4(c) 可以看到,整个场景得到了完整的重建。

## 3 结束语

该文提出了一种改进的 RGB-D SLAM 算法来进行即时定位与地图构建。在 SLAM 前端,该文采用 ORB 特征提取算法来进行特征点的提取,使其可以运行在嵌入式设备和缺乏 GPU 的设备上;采用 BoW 模型改进了匹配方法,极大地减少了匹配所需要的时间,提高了算法的实时性;利用 PROSAC 算法结合 PnP 算法计算相机位姿,提高了精度的同时也节省了时间,并且利用非线性优化进一步提高了位姿的精度;在传统的帧选取方法上加入了冗余帧删除机制,便于帧的管理和维护。在 SLAM 后端,该文采用了 BoW 模型进行闭环检测,利用结构一致性提高了闭环的准确率,增强了算法的鲁棒性;通过通用图优化工具 g2o 对位姿图进行优化,得到了全局一致的位姿。该文采用贪心三角化算法来构建三维网格地图。最后,该文选取 TUM 数据集里的 Fr1 数据包进行实验,并通过与文献[19]、文献[24]和文献[25]进行了实时性和准确性的对比,表明了该算法的优越性。在下一步的工作中,将会结合 IMU(惯性测量单元)等高精度设备更进一步优化相机位姿,利用深度学习方法提取语义信息,构建语义 SLAM。

### 参考文献:

- [1] 陈世浪,吴俊君. 基于 RGB-D 相机的 SLAM 技术研究综述[J]. 计算机工程与应用,2019,55(7):30-39.
- [2] 周彦,李雅芳,王冬丽,等. 视觉同时定位与地图创建综述[J]. 智能系统学报,2018,13(1):97-106.
- [3] TAKETOMI T, UCHIYAMA H, IKEDA S. Visual SLAM algorithms: a survey from 2010 to 2016[J]. IPSJ Transactions on Computer Vision and Applications,2017,9:16.

- [4] CADENA C, CARLONE L, CARRILLO H, et al. Past, present, and future of simultaneous localization and mapping: toward the robust-perception age[J]. *IEEE Transactions on Robotics*, 2016, 32(6): 1309–1332.
- [5] 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.
- [6] MIHALYI R G, PATHAK K, VASKEVICIUS N, et al. Robust 3D object modeling with a low-cost RGBD-sensor and AR-markers for applications with untrained end-users[J]. *Robotics and Autonomous Systems*, 2015, 66: 1–17.
- [7] 高成强, 张云洲, 王晓哲, 等. 面向室内动态环境的半直接法 RGB-D SLAM 算法[J]. *机器人*, 2019, 41(3): 372–383.
- [8] 谷晓琳, 杨 敏, 张 焱, 等. 一种基于半直接视觉里程计的 RGB-D SLAM 算法[J]. *机器人*, 2020, 42(1): 39–48.
- [9] HENRY P, KRAININ M, HERBST E, et al. RGB-D mapping: using depth cameras for dense 3D modeling of indoor environments[M]//*Experimental robotics*. Berlin, Heidelberg: Springer, 2010: 22–25.
- [10] HENRY P, KRAININ M, HERBST E, et al. RGB-D mapping: using kinect-style depth cameras for dense 3D modeling of indoor environments[J]. *International Journal of Robotics Research*, 2012, 31(5): 647–663.
- [11] FISHLER M, BOLLES R. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography[J]. *Communications of the ACM*, 1981, 24(6): 381–395.
- [12] NEWCOMBE R A, IZADI S, HILLIGES O, et al. KinectFusion: real-time dense surface mapping and tracking[C]//*The 10th IEEE international symposium mixed and augmented reality*. Basel, Switzerland: IEEE, 2011: 127–136.
- [13] WHELAN T, KAESSE M, JOHANSSON H, et al. Real-time large-scale dense RGB-D SLAM with volumetric fusion[J]. *International Journal of Robotics Research*, 2014, 34(4–5): 598–626.
- [14] ENDRES F, HESS J, STURM J, et al. 3-D mapping with an RGB-D camera[J]. *IEEE Transactions on Robotics*, 2014, 30(1): 177–187.
- [15] LOWE D G. Distinctive image features from scale-invariant keypoints[J]. *International Journal of Computer Vision*, 2004, 60(2): 91–110.
- [16] BAY H, TUYTELAARS T, GOOL L V. SURF: speeded up robust features[J]. *Computer Vision & Image Understanding*, 2006, 110(3): 404–417.
- [17] RUBLEE E, RABAUDE V, KONOLIGE K, et al. ORB: an efficient alternative to SIFT or SURF[C]//*IEEE international conference on computer vision*. Barcelona, Spain: IEEE, 2011: 2564–2571.
- [18] MUR-ARTAL R, TARDOS J D. ORB-SLAM2: an open-source slam system for monocular, stereo, and RGB-D cameras[J]. *IEEE Transactions on Robotics*, 2017, 33(5): 1255–1262.
- [19] 张 震, 郑 宏, 周 璇, 等. 一种结合 ORB 特征和视觉词典的 RGB-D SLAM 算法[J]. *计算机工程与应用*, 2018, 54(12): 170–176.
- [20] CHUM O, MATAS J. Matching with PROSAC – progressive sample consensus[C]//*IEEE computer society conference on computer vision and pattern recognition*. San Diego: IEEE, 2005: 220–226.
- [21] GALVEZ-LÓPEZ 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.
- [22] HARTLEY R, ZISSERMAN A. 计算机视觉中的多视图几何[M]. 韦 穗, 杨尚骏, 译. 合肥: 安徽大学出版社, 2002.
- [23] STURM J, ENGELHARD N, ENDRES F, et al. A benchmark for the evaluation of RGB-D SLAM systems[C]//*IEEE/RSJ international conference on intelligent robots and systems*. Vilamoura-Algarve, Portugal: IEEE, 2012: 573–580.
- [24] ENDRES F, HESS J, ENGELHARD N, et al. An evaluation of the RGB-D SLAM system[C]//*IEEE international conference on robotics and automation*. Saint Paul, USA: IEEE, 2012: 1691–1696.
- [25] 付梦印, 吕宪伟, 刘 彤, 等. 基于 RGB-D 数据的实时 SLAM 算法[J]. *机器人*, 2015, 37(6): 683–692.