

DOI:10.19651/j.cnki.emt.2209624

改进 ORB-SLAM2 算法的关键帧选取及 地图构建研究^{*}

徐武¹ 高寒¹ 王欣达¹ 张强¹ 秦浩然² 吴庆畅³

(1. 云南民族大学电气信息工程学院 昆明 650031; 2. 中国石油运输公司 乌鲁木齐 830014;

3. 云南民族大学澜湄职业技术学院 昆明 650031)

摘要: 针对传统 ORB-SLAM2 算法精度较低、图片帧跟踪容易丢失的问题,以及没有构建稠密点云地图与 OctoMap 的问题,原本构建的稀疏点云地图不能直接应用于机器人的三维路径规划。在传统 ORB-SLAM2 算法基础上,对关键帧的选取进行了改进。首先,在传统的 ORB-SLAM2 算法的基础上,在相邻图像帧间添加相对运动量的综合变换因数,并添加帧间特征点跟踪来提高关键帧选取的准确性;然后,使用已选取关键帧进行稠密点云地图与 OctoMap 的构建;最后,在 TUM 数据集上进行验证,并基于真实环境进行物理测试。实验结果表明,改进的关键帧选取方法在保证关键帧选取的准确性和快速性的前提下,可以增加 ORB-SLAM2 算法定位精度,有效的缓解了图片帧跟踪易丢失问题,并且构建的八叉树地图可直接用于机器人的三维路径规划。

关键词: ORB-SLAM2 算法;关键帧选取;相对运动量;八叉树地图

中图分类号: TP391.41;TP242.6 **文献标识码:** A **国家标准学科分类代码:** 520.6

Research on key frame selection and map construction of improved ORB-SLAM2 algorithm

Xu Wu¹ Gao Han¹ Wang Xinda¹ Zhang Qiang¹ Qin Haoran² Wu Qingchang³

(1. College of Communication and Electronic Engineering, Yunnan Minzu University, Kunming 650031, China;

2. China Petroleum Transportation Corporation, Urumqi 830014, China;

3. Lanmei Vocational and Technical College, Yunnan Minzu University, Kunming 650031, China)

Abstract: Aiming at the problems of low accuracy of traditional ORB-SLAM2 algorithm, easy loss of picture frame tracking, and the lack of dense point cloud map and octomap, the originally constructed sparse point cloud map can not be directly applied to the three-dimensional path planning of robot. Based on the traditional ORB-SLAM2 algorithm, this paper improves the selection of key frames. Firstly, based on the traditional ORB-SLAM2 algorithm, the comprehensive transformation factor of relative motion is added between adjacent image frames, and the inter frame feature point tracking is added to improve the accuracy of key frame selection. Then, the selected key frame is used to construct the dense point cloud map and octomap. Finally, the verification is carried out on the tum data set, and the physical test is carried out based on the real environment. The experimental results show that the improved key frame selection method can increase the positioning accuracy of ORB-SLAM2 algorithm on the premise of ensuring the accuracy and rapidity of key frame selection, effectively alleviate the problem of easy loss of picture frame tracking, and the octree map can be directly used for robot 3D path planning.

Keywords: ORB-SLAM2 algorithm; key frame selection; relative movement; octree map

0 引言

近年来,同时定位与地图构建(simultaneous localization

and mapping, SLAM)在机器人、无人机、增强现实(augmented reality, AR)等方面得到越来越广泛的应用^[1-2],它指的是机器人在未知环境中运动时增量地构建周

收稿日期:2022-04-15

^{*} 基金项目:国家自然科学基金(U1802271)、少数民族优秀文化保护传承工程项目(2021YNMW005)资助

围环境地图,同时运用已构建好的地图来评估机器人自身位姿^[3]。SLAM 可以按其所载的传感器类型划分为两类。第 1 类激光 SLAM^[4]大多应用雷达传感器,虽然基于雷达的 SLAM 精度较高,但是雷达价格昂贵且体积较大。而第 2 类视觉 SLAM 应用摄像头作为传感器,摄像头相对于雷达来说具有价格便宜、重量更轻、功耗更低等优点,因此使得性价比较高的视觉 SLAM 逐渐成为机器人实现自主导航的主流方法^[5-6]。

视觉 SLAM 经过了 30 年的发展,出现了一些较为经典的 SLAM 系统如 LSD-SLAM、ORB-SLAM、RGB-D SLAM 等。文献[7]提出利用单目相机实现的 LSD-SLAM,其实现了使用直接法构建半稠密的地图。文献[8]首次提出了 ORB-SLAM 算法,算法由跟踪线程、局部地图构建线程、回环检测线程三线程组成,使用的是基于 ORB 特征计算的特征点法,具有良好的实时性。文献[9]在 ORB-SLAM 的基础上提出了 ORB-SLAM2,该算法支持单目、双目、RGB-D 三种相机,使用光速平差法进行后端优化,增加了定位精度。文献[10]提出了基于深度相机的 RGB-D SLAM 算法,其实现了使用深度相机获取图像信息来构建稠密地图。文献[11]提出了 VINS-Mono,此方法在单目惯导视觉里程计上具有较高的精度。

相较于其他的视觉 SLAM 算法,ORB-SLAM2 算法具有定位精确度高、支持多种相机模式、系统框架清晰等优点,受到国内外广大研究者的青睐。在传统 ORB-SLAM2 算法中,常用的方法是使用关键帧进行定位与地图构建,关键帧是提高算法精确度、降低系统内存消耗的重要方法。文献[12]使用 PROSAC 算法代替了 RANSAC 算法,提高了剔除误匹配的效率。可以看到,针对 ORB-SLAM2 关键帧的选取已经有学者进行了探究并取出大量成果,但是还是存在着定位容易丢失、精度不高、误差较大的问题。

为了解决以上问题,本文在 ORB-SLAM2 基础上,对关键帧的选取进行了改进。首先,在传统的 ORB-SLAM2 算法的基础上,在相邻图像帧间添加相对运动量的综合变换因数(包含转角变换因数和综合平移变换因数,且二者没有紧密关联),同时添加帧间特征点跟踪来增加关键帧选取的准确性。另一方面,ORB-SLAM2 所建立的稀疏点云地图无法用于机器人自主导航^[13],因此,本文在原有算法三线程的基础上添加构建稠密点云地图与 OctoMap 的线程^[14-15],得到的 OctoMap 可直接应用于机器人的自主导航和三维路径规划。最后,在 TUM 数据集^[16]上进行实验验证,并且就真实环境进行物理测试,进一步对比改进关键帧选取后的定位精度。

1 系统框架

ORB-SLAM2 包括跟踪、局部地图构建、回环检测 3 个线程,并结合全局 BA(bundle adjustment)优化线程实现整体系统功能。本文在 ORB-SLAM2 算法的基础上增加相

对运动量的旋转和综合平移量,结合帧间特征点跟踪提高关键帧选取的准确性,并添加构建稠密点云地图和八叉树地图线程。本文的整体系统框架图如图 1 所示。

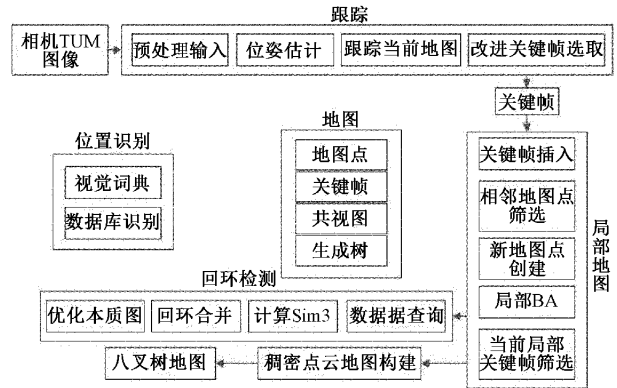


图 1 整体系统框架

整体系统线程主要分为如下部分:

1)跟踪线程,这一线程的工作是从 TUM 图像中提取特征点,与前一帧图像的特征点相匹配或是通过重定位进行位姿初始化,使用 RANSAC 算法剔除误匹配,对当前帧位姿进行优化,最后在相邻图像帧间添加相对运动量的转角变换因数和综合平移变换因数,并结合帧间特征点跟踪选取新的关键帧。

2)构建局部地图线程,这一线程接收跟踪线程产生的关键帧并插入到地图中,对生成的相邻地图点进行筛选,并创建新地图点;通过局部 BA(local BA)进行优化,最后过滤局部关键帧,剔除冗余的关键帧。

3)回环检测线程,这一线程主要分为回环判断和回环校正两个部分。回环判断是先使用 DBow 数据库进行检测,当检测到存在闭环关键帧时,通过 Sim3 算法计算相似变换。回环校正主要包括回环合并和全局的位姿优化。

4)稠密点云地图和 OctoMap 构建线程,这一线程利用 ORB-SLAM2 算法基础上进行稠密点云地图和 OctoMap 的构建。

2 改进关键帧选取算法

2.1 传统 ORB-SLAM2 关键帧选取算法

首先,传统 ORB-SLAM2 算法要求当前帧的内点数必须大于设置的最小阈值,而且重叠率不能太大。在满足以上条件时,判断下列情形,若有一项符合,则插入关键帧。

1)避免跟踪丢失,在距离上次插入关键帧已经过去最大帧时,进行关键帧的插入。(最大帧为预设定的最大阈值)。

2)避免图像的重叠率过高,并且局部建图线程正处于空闲的状态,在距离上次插入关键帧已经过去最小帧后,进行关键帧的插入。(最小帧为预设定的最小阈值)。

3)由于局部建图线程的计算能力有限,且存在信息冗余,因此必须确保关键帧队列中的关键帧数量不能超过 3 个。

ORB-SLAM2 的关键帧选取算法只与系统关键帧的通过数量有关,在某些摄像机出现转动或抖动时,仍然会出现关键帧信息收集不足的问题,导致位姿出现较大误差,甚至出现关键帧丢失或者跟踪丢失。本文在相邻图像帧间添加相对运动量的转角变换因数和综合平移变换因数,并添加帧间特征点跟踪来增加关键帧选取的准确性。

2.2 ORB-SLAM2 关键帧选取算法的改进

为了提高 ORB-SLAM2 算法准确度,在传统的 ORB-SLAM2 算法的基础上,在相邻图像帧间添加相对运动量的转角变换因数和综合平移变换因数,并添加特征点跟踪作为另一个判定条件,两者结合共同增加关键帧选取的准确性。首先对相对运动量的旋转和平移进行计算:

假设得到两组匹配的特征点为 $p = \{p_1, p_2, \dots, p_n\}$ 和 $p' = \{p'_1, p'_2, \dots, p'_n\}$,通过构建最小二乘方法,求解 (R, t) ,如下式:

$$\operatorname{argmin}_{R,t} \sum_{i=1}^n \| p'_i - (R \cdot p_i + t) \|^2 \quad (1)$$

对每个点进行去质心计算:

$$\begin{aligned} q_i &= p_i - p \\ q'_i &= p'_i - p' \end{aligned} \quad (2)$$

代入式(2)对式(1)进行化简得:

$$\operatorname{argmin}_{R,t} \sum_{i=1}^n \{ \| q'_i - R \cdot q_i \|^2 + \| p'_i - R \cdot p_i - t \|^2 \} \quad (3)$$

对式(3)第 1 个累加和展开化简:

$$\begin{aligned} \operatorname{argmin}_R \sum_{i=1}^n \{ \| q'_i - R \cdot q_i \|^2 = \\ \operatorname{argmin}_R \sum_{i=1}^n (q'_i{}^T q'_i - 2q'_i{}^T R q_i + q'_i{}^T R^T R q_i) \end{aligned} \quad (4)$$

对上式进行化简,去除 R 的无关项,优化后的函数为:

$$\operatorname{argmin}_R \sum_{i=1}^n -q'_i{}^T R q_i = \operatorname{argmin}(-\operatorname{tr}(R \sum_{i=1}^n q'_i{}^T q_i)) \quad (5)$$

令 $M = \sum_{i=1}^n q'_i{}^T q_i$,然后利用 SVD 分解得到:

$$\operatorname{SVD}(M) = USV^T \quad (6)$$

从而求的:

$$R = UV^T \quad (7)$$

通过 R 求 t :

$$t = p' - R p \quad (8)$$

根据计算得出的 R 和 t ,求出帧间的相对运动量 D ,并在 ORB-SLAM2 系统预先设定最大阈值 D_{\max} 与最小阈值 D_{\min} :

$$D = \|\Delta t\| + \min(2\pi - \|R\|, \|R\|) \quad (9)$$

本文在式(9)的基础上添加了综合平移变换因数和转角变换因数,如式(10)所示。

$$D = (\alpha + \gamma) \|\Delta t\| + \beta \min(2\pi - \|R\|, \|R\|) \quad (10)$$

其中, $(\alpha + \gamma)$ 为综合平移变换因数,其中, α 为相对于

相机视角的左右平移变换因数(当相机没有发生左右平移时,该值为 0), γ 为相对于相机视角的前后平移变换因数(当相机没有发生前后平移时,该值为 0); β 为转角变换因数, β 的大小随转角变化呈现指数变化(转角变大则 β 变大),且 $(\alpha + \gamma)$ 和 β 无紧密相关性, $(\alpha + \gamma)$ 的值不随 β 的变化而变化。在帧间相对运动量中,转角作为主要变化量,平移作为次要变化量,当出现转角变化远大于位移变化时, $(\alpha + \gamma)$ 忽略不计,由 $\beta(2\pi - \|R\|, \|R\|)$ 决定相对运动量,取值范围为 $0 \sim 90^\circ$,即当转角度数超过 90° 时,相机无法继续跟踪。

对求出的 D 与以设定的阈值进行比较:

$D < D_{\min}$ 时, $Frame_{cur} \neq Frame_{key}$;

$D_{\min} \leq D \leq D_{\max}$ 时, $Frame_{cur} = Frame_{key}$;

$D_{\max} < D$ 时,同时转角角度不超过 90° 时,

$Frame_{cur} = Frame_{key}$ 。

式中: $Frame_{cur}$ 为当前帧, $Frame_{key}$ 为待选关键帧。

通过上面的判定后,将当前帧设为待选关键帧,然后继续进行下面的判定:

$\operatorname{Track}(Frame_{key-1}, Frame_{key}) > \gamma$

式中: Track 为计算当前待选关键帧和上一帧之间跟踪到特征点数量, γ 为可调参数。满足上面的判定,说明帧间的匹配度高、相关性强,将当前待选关键帧设定为关键帧,并插入关键帧。

系统的算法流程如图 2 所示。系统首先判定是否可以插入关键帧,即不处于定位模式中和全局闭环中,以及距离上一次重定位较远时。

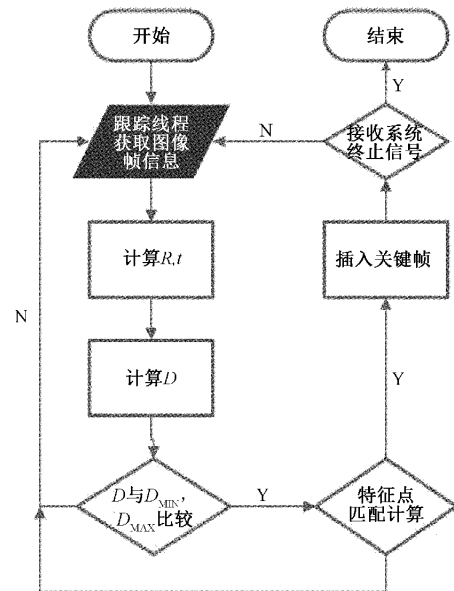


图 2 算法流程

然后利用上面公式计算 (R, t) 与相对运动量 D ,使用 D 与已设定的阈值 D_{\min} 和 D_{\max} 比较,如果当前帧的相对运动量大于设定的阈值,说明相机对于前帧发生了较大的

相对运动,此时需要生成新的关键帧,确保定位不会出现较大的误差,不会出现跟踪丢失。所以加入相对运动量有助于减少定位误差,增加系统的稳定性。

最后结合帧间特征点跟踪方法进一步筛选关键帧,对关键帧的选取进一步优化。整个系统不断循环,直至结束。

3 稠密点云地图与 OctoMap

ORB-SLAM2 系统是偏于定位的系统,构建了稀疏点云地图,所以本文在 ORB-SLAM2 系统的基础上添加了稠密点云地图和 OctoMap 的构建线程。

1) 稠密点云地图的构建

利用 RGB-D 图像进行稠密点云地图的构建,首先根据相机模型成像原理得:

$$s \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = C \cdot (R \cdot x_i + t) \quad (11)$$

其中, $X = \{x_1, x_2, x_i, \dots, x_n\}$ 为空间中点的集合, $x_i = [x, y, z]$ 为空间点的三维坐标, C 为相机内参, R 为旋转矩阵, t 为平移矩阵, s 为比例关系因子。

其次利用计算深度信息来得到点云数据:

$$\begin{bmatrix} m_x \\ m_y \\ m_z \end{bmatrix} = \begin{bmatrix} (n - c_x) \cdot m_z / f_x \\ (n - c_y) \cdot m_z / f_y \\ d \end{bmatrix} \quad (12)$$

其中, $m = [m_x, m_y, m_z]$ 为点云数据点, d 为深度图像中点的深度值, c_x, c_y, f_x, f_y 为相机内参。

根据上述公式,利用计算得到的图像深度信息,以得到稠密点云地图。并且稠密点云地图的储存量太大,所以在稠密点云地图的基础上进行 OctoMap 的构建。

2) OctoMap 的构建

OctoMap 可以表示为一个大的正方体,根节点为最大的正方体。在根节点的下方有 8 个结点,每一个结点往下分都会被成 8 个结点,然后继续向下延伸,直到叶子为止。

每个小正方体都有可能被占据或不被占据。通常情况下,用 0 表示节点未被占据,1 表示节点已被占据。0.5 表示节点不确定是否占据,越大则表示被占据的可能性越高,反之亦然。图 3 为八叉树原理图。

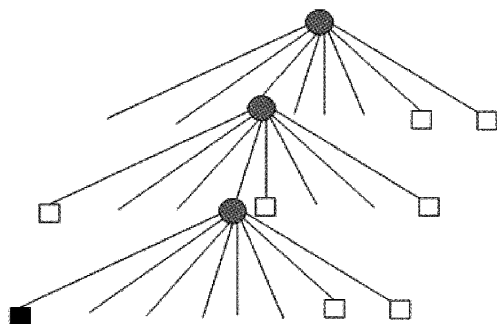


图 3 八叉树原理

在八叉树中可以用概率来表示叶子被占用的概率,但是概率表示的形式可能超出 $[0, 1]$ 这个区间,所以通常不直接使用概率表示某个节点被占据,而是用对数几率 (log-odds) 来代替概率描述。对数几率 (log-odds) 又称 logit 函数,公式如下:

$$L = \text{logit}(p) = \log \frac{p}{1-p} \quad (13)$$

其反变换为:

$$p = \text{logit}^{-1}(L) = \frac{\exp(L)}{\exp(L) + 1} \quad (14)$$

这样当 $L \in (-\infty, \infty)$ 时, $p \in (0, 1)$, p 不会超出 $(0, 1)$ 区间。用 $L(n | z_{1:T})$ 表示叶子结点被占据概率,可将上式表示如下:

$$L(n | z_{1:T}) = L(n | z_{1:T-1}) + L(n | z_T) \quad (15)$$

这样就可以对八叉树地图进行即时更新,实现八叉树地图的实时构建。

4 实验结果与分析

所有实验均在 Ubuntu18.04 环境下进行,所用计算机 CPU 为 2.30 GHz Intel i7,运行内存为 16 GB。实验数据集来自 TUM RGB-D Benchmark,取其中 fr1_desk、fr2_xyz 和 fr1_room 序列。实验比较了 ORB-SLAM2 关键帧选取算法与改进后的关键帧选取算法对 ORB-SLAM2 实时性、定位精度的影响,并构建了稠密点云地图与八叉树地图。

4.1 ORB-SLAM2 实时性分析

对于任何 SLAM,实时性都是很重要评估指标,表 1 记录了在不同数据集下 ORB-SLAM2 的平均跟踪时间,实验均进行 3 组,取其平均值。

表 1 平均跟踪时间比较 ms

数据集	ORB-SLAM2 关键帧选取算法	改进的关键帧选取算法
fr1_desk	16.14	17.25
fr2_xyz	14.46	14.58
fr1_room	13.77	14.36

从表 1 中可以看出,改进后的关键帧选取算法在增加相对运动量与帧间特征点跟踪后平均跟踪时间较原算法有增加,但是相差不是很大,所以两种算法实时性相差较小。

4.2 改进的 ORB-SLAM2 算法定位准确度对比

图 4(a)为 TUM 数据集在 fr1_desk 场景下的轨迹与 ORB-SLAM2 关键帧选取算法在 fr1_desk 场景下的轨迹对比图。图 4(b)为 fr1_desk 场景下真实且连续的轨迹与改进的关键帧选取算法在 fr1_desk 场景下的轨迹对比图。可以看出,在图 4(b)中关键帧选取算法在 fr1_desk 场景下的轨迹奇点较少,与真实轨迹更加接近。

对于定位精度,可以采用绝对位姿误差 ATE 比较,当

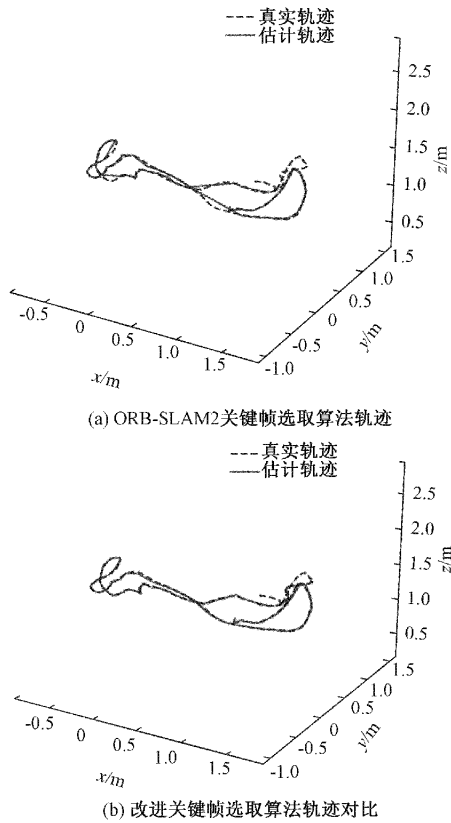


图 4 两种关键帧选取算法的轨迹对比

绝对位姿误差越小时,算法定位精度越高。计算公式如下所示:

$$F_i = Q_i^{-1} S P_i \quad (16)$$

式中: F_i 是第 i 帧 ATE, P_i 是算法估计位姿, Q_i 是真实位姿, S 是真实位姿与估计位姿之间的相似旋转矩阵。

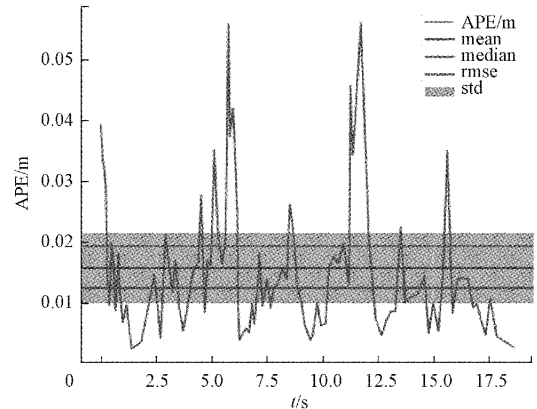
把算法估计轨迹与真实轨迹放到一起,绘制绝对位姿误差图,如图 5 所示。

图 5 为两种算法的绝对位姿误差对比图,其中图 5(a) 为传统 ORB-SLAM2 关键帧选取算法绝对位姿误差图,图 5(b) 为改进关键帧选取算法后得到的绝对位姿误差图,从两张图片中看出相较于 ORB-SLAM2 关键帧选取算法,改进后的关键帧选取算法有了较大的提升。利用绝对轨迹误差进行比较,表 2 为两种算法在 3 个不同数据集下的最大轨迹误差。

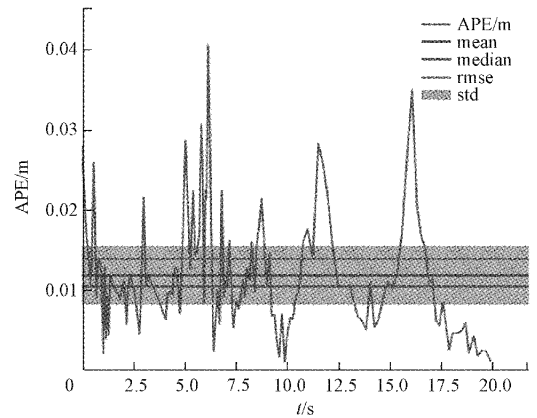
表 2 最大轨迹误差对比

数据集	ORB-SLAM2 关键帧选取算法	改进的关键帧选取算法
fr1_desk	0.057	0.041
fr2_xyz	0.014	0.010
fr1_room	0.183	0.132

由表 2 可以看出,改进关键帧选择算法与传统的 ORB-



(a) ORB-SLAM2 关键帧选取算法绝对位姿误差图



(b) 改进关键帧选取算法绝对位姿误差图

图 5 两种算法的绝对位姿误差图

SLAM2 关键帧选取算法相比,在 fr1_desk、fr2_xyz 和 fr1_room 数据集下,优化后的算法误差相较于原算法分别减少了 28.0%、28.6%、27.8%。

4.3 ORB-SLAM2 关键帧数量分析

本文将传统的 ORB-SLAM2 关键帧选取算法和改进的 ORB-SLAM2 的关键帧选取算法分别在 fr1_desk、fr2_xyz 和 fr1_room 数据集下所获得的关键帧的数量进行了对比,如图 6 所示。由图可知,与原始方法比较,改进后的关键帧选择方法得到的关键帧数目更多,说明改进后的关键帧选取算法可以获得更多的关键帧以提高定位精度与系统稳定性。

4.4 稠密点云地图与 OctoMap 的构建

原 ORB-SLAM2 系统是偏于定位的系统,它构建了稀疏点云地图,但稀疏点云无法应用于机器人的导航和三维路径规划。基于以上问题,本文在原系统基础上构建了稠密点云地图和 OctoMap,并利用机器人实时未知环境探索验证了算法的可行性。

图 7(a) 和 (c) 为在传统的 ORB-SLAM2 算法在 fr1_desk 数据集下构建的稠密点云地图与 OctoMap。

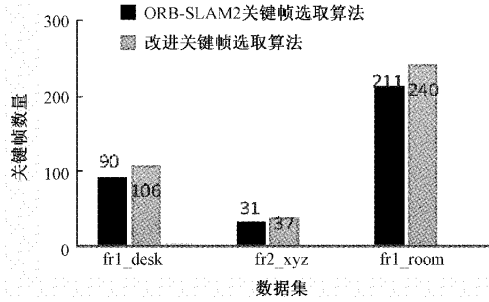
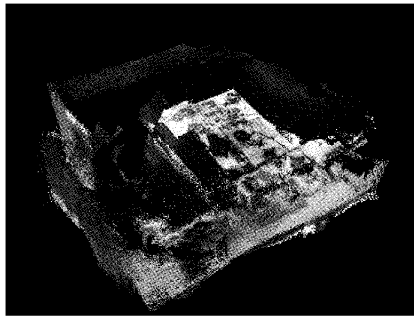
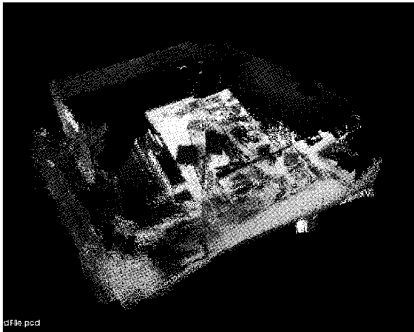


图 6 不同数据集下两种算法获得的关键帧数量



(a) 稠密点云地图



(b) 改进关键帧选取算法后生成的稠密点云地图



(c) OctoMap



(d) 改进关键帧选取算法后生成的OctoMap

图 7 稠密点云地图与 OctoMap

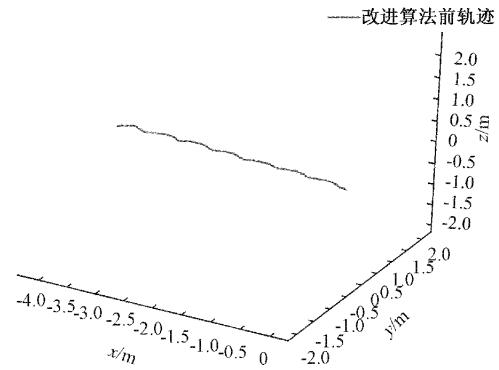
从图 7(a)中可以看到,构建的稠密点云地图保留了很多细节的方面,如桌子、椅子、电脑等,造成了信息冗余,增大了地图储存的压力。图 7(c)中的 OctoMap 在保留必要的信息的基础上减少了信息冗余,地图储存压力减小,有助于增加算法运行的实时性。

图 7(b)和(d)分别为改进关键帧选取算法后生成的稠密点云地图和八叉树地图,通过对比可以看出,相较于传统的 ORB-SLAM2 算法,改进后的算法生成的稠密点云地图点云数量更多,地图完整度更高;而 OctoMap 中,地图完整度更高。

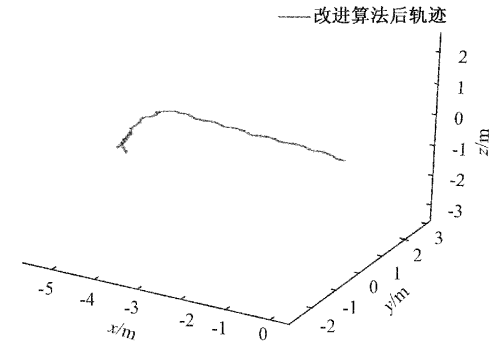
4.5 真实环境物理测试

为验证 ORB-SLAM2 算法中改进关键帧在实际应用中的效果,本文对真实环境进行物理测试,对比改进关键帧选取前和改进后的定位精度,同时,生成稠密点云地图和八叉树地图,可用做机器人的实时路径规划。该实验使用 Intel[®] RealSense™ Depth Camera D435i 深度相机采集实验数据。

图 8(a)和(b)是改进关键帧选取前后在同一真实场景下的轨迹图。从图 8(b)可以看出,在拐角处,当遇到特征点较少的情况下(例如:白色墙体)还可以继续跟踪,不会出现特征点丢失的情况;由图 8(a)所示,未改进关键帧选取前,一旦遇到特征点较少的情况,跟踪极易中断,无法继续进行跟踪。



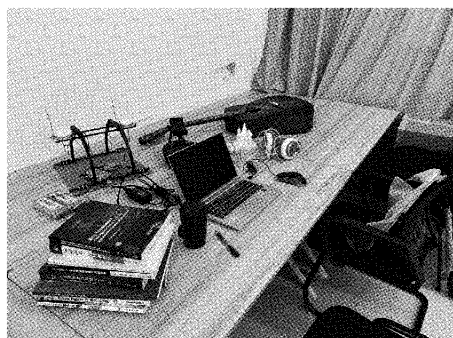
(a) 改进关键帧选取前的物理测试轨迹



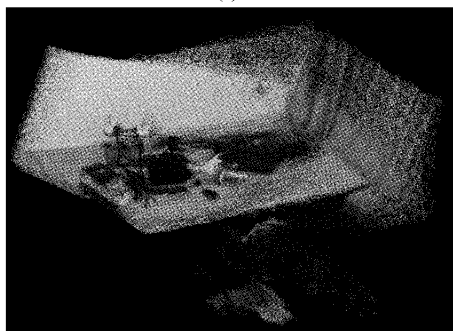
(b) 改进关键帧选取后的物理测试轨迹

图 8 物理测试轨迹对比

图 9 为改进关键帧选取算法后在真实环境下构建的稠密点云地图和 OctoMap。



(a) 原图



(b) 稠密点云地图



(c) OctoMap

图 9 改进关键帧选取算法后生成的地图

5 结 论

本文提出了一种改进关键帧选取的 ORB-SLAM2 算法,在传统的 ORB-SLAM2 算法的基础上,在相邻图像帧间添加相对运动量的转角变换因数和综合平移变换因数,且二者无紧密关联,有效的避免了跟踪易丢失的问题,在 fr1_desk、fr2_xyz 和 fr1_room 数据集下,优化后的算法误差相较于原算法分别减少了 28.0%、28.6%、27.8%。提高了系统定位精度。此外,本文在真实环境中进行物理测试,进一步验证了改进 ORB-SLAM2 算法的关键帧选取后的定位精确度有所提高,跟踪不易丢失。同时在 ORB-SLAM2 算法的基础上添加了构建稠密点云图和 OctoMap,与原本构建的稀疏点云地图不同的是,稠密点云图和 OctoMap 可以直接应用于机器人的导航与三维路线规划。

参考文献

- [1] BELTER D, NOWICKI M R. Optimization-based legged odometry and sensor fusion for legged robot continuous localization[J]. Robotics and Autonomous Systems, 2019, 111: 110-124.
- [2] ANGLADON V, GASPARINI S, CHARVILLAT V, et al. An evaluation of real-time RGB-D visual odometry algorithms on mobile devices[J]. Journal of Real-Time Image Processing, 2017: 1-18.
- [3] 孙龙龙, 江明, 焦传佳. 基于运动矢量的改进视觉 SLAM 算法[J]. 电子测量与仪器学报, 2020, 34(9): 23-31.
- [4] 周治国, 曹江微, 邸顺帆. 3D 激光雷达 SLAM 算法综述[J]. 仪器仪表学报, 2021, 42(9): 13-27.
- [5] 罗会兰, 陈鸿坤. 基于深度学习的目标检测研究综述[J]. 电子学报, 2020, 48(6): 1230-1239.
- [6] 陆兴鹏, 王明羽, 曹扬, 等. 一种基于帧图像的动态视觉传感器样本集建模方法[J]. 电子学报, 2020, 48(8): 1457-1464.
- [7] ENGEL J, SCH T, CREMERS D. LSD-SLAM: Large-scale direct monocular SLAM [C]. European Conference on Computer Vision Springer, Cham, 2014: 834-849.
- [8] MUR-ARTAL R, MOTIEL J M, TARDOS J D. ORB-SLAM: A versatile and accurate monocular SLAM system[J]. IEEE Transactions on Robotics, 2015, 31(5): 1147-1163.
- [9] MUR-ARTAL R, MOTIEL J M, 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.
- [10] ENDRES F, HESS J, STURM J, et al. 3D mapping with an RGB-D camera [J]. IEEE Transactions on Robotics, 2013, 30(1): 177-187.
- [11] QIN T, LI P, SHEN S. Vins-mono: A robust and versatile monocular visual-inertial state estimator [J]. IEEE Transactions on Robotics, 2018, 34(4): 1004-1020.
- [12] 席志红, 王洪旭, 韩双全. 基于 ORB-SLAM2 系统的快速匹配剔除算法与地图构建[J]. 计算机应用, 2020, 40(11): 3289-3294.
- [13] 陆建伟. 基于视觉的 ORB-ISM 室内同时定位与建图研究[D]. 太原: 太原理工大学, 2019: 37-48.
- [14] 李雅芳. 基于 RGB-D 的稠密 ORB-SLAM 研究[D]. 湘潭: 湘潭大学, 2018: 22-50.
- [15] 席志红, 韩双全, 王洪旭. 基于语义分割的室内动态场景同步定位与语义建图[J]. 计算机应用, 2019, 39(10): 2847-2851.

[16] 宋玉琴,熊高强,曾贺东,等.多平面点优化的单目 SLAM 方法[J].国外电子测量技术,2021,40(10):40-45.

作者简介

徐武,工学硕士,教授,主要研究方向为虚拟现实技术、图像处理与识别等。

E-mail:1369925132@qq.com

高寒,硕士研究生,主要研究方向为图像处理、机器视觉等。

E-mail:1831552634@qq.com

吴庆畅(通信作者),硕士,副教授,主要研究方向为图像处理、机器视觉等。

E-mail:57711418@qq.com