

基于 ORB-SLAM2 的改进特征匹配与稠密地图算法^{*}

王喜红¹ 雷斌^{1,2} 李园园¹ 张黎¹ 李德仓^{1,2}

(1. 兰州交通大学机电技术研究所 兰州 730070; 2. 甘肃省物流及运输装备信息化工程技术研究中心 兰州 730070)

摘要: 针对 ORB-SLAM2 算法在特征匹配时容易出现误匹配以及无法构建稠密地图的问题,引入 GMS 算法来改进 ORB-SLAM2 算法中的误匹配问题以及添加稠密地图线程。首先,建立图像金字塔,对每层图像金字塔进行网格划分,提取特征点,对每个网格中的特征点引入四叉树策略进行筛选,得到均匀的特征点;其次,在特征匹配阶段引入 GMS 算法剔除误匹配;最后,根据位姿估计和关键帧构建稠密点云地图。通过 TUM 数据集上的实验验证,结果表明改进算法的匹配数比原 ORB-SLAM2 算法增加了 18.36%,匹配用时减少了 8.53%,将改进算法应用于移动机器人自动导航和避障中,能够提高系统的可靠性和运行效率。

关键词: 特征匹配;GMS 算法;剔除误匹配;稠密点云地图

中图分类号: TP242; TN98 文献标识码: A 国家标准学科分类代码: 510.4050

Improved feature matching and dense mapping algorithm based on ORB-SLAM2

Wang Xihong¹ Lei Bin^{1,2} Li Yuanyuan¹ Zhang Li¹ Li Decang^{1,2}

(1. Mechatronics T&R Institute, Lanzhou Jiaotong University, Lanzhou 730070, China; 2. Gansu Provincial Engineering Technology Center for Informatization of Logistics and Transport Equipment, Lanzhou 730070, China)

Abstract: To address problem that the ORB-SLAM2 algorithm is prone to mismatching and cannot build a dense map during feature matching, the GMS algorithm is introduced to improve the mismatching problem in the ORB-SLAM2 algorithm and add a dense map thread. First, an image pyramid is established, and a grid division is performed on each layer of the image pyramid to extract feature points. A four-tree strategy is introduced for feature point selection in each grid, resulting in a uniform distribution of feature points. Second, the GMS algorithm is introduced in the feature matching stage to eliminate false matches. Finally, the dense point cloud map is built based on the pose estimation and key frames. Through the experimental verification on TUM data set, the results show that the matching number of the improved algorithm is 7.82% higher than that of the original ORB-SLAM2 algorithm, and the matching time is reduced by 8.53%. The improved algorithm is applied to the automatic navigation and obstacle avoidance of mobile robot, which can improve the reliability and operation efficiency of the system.

Keywords: feature matching; GMS algorithm; eliminate mismatching; map of dense point clouds

0 引言

在未知的环境中,机器人根据传感器测量到的数据同时估计自身的位姿和周围的环境地图,这个问题被称为同时定位与地图构建(simultaneous localization and mapping,SLAM)^[1]。SLAM 技术可以获取精确的位置信息与地图环境信息^[2-3],在机器人、无人机、增强现实(AR)等各种邻域起着重要作用^[4]。

根据 SLAM 技术传感器的不同,主要分为激光 SLAM

和视觉 SLAM 两种类型,视觉 SLAM 相比于激光 SLAM 更具有优势,是当前 SLAM 传感器研究热点之一^[5]。ORB-SLAM2^[6]是一种基于单目、双目和 RGB-D 相机的实时视觉 SLAM 方案,可以构建三维地图并定位相机的位姿^[7]。谷学静等^[8]提出一种基于四叉树分解法的 ORB 算法,对待处理的原始图像经过四叉树分解法来剔除掉灰度变化不明显的区域,再进行 FAST 特征点检测,在特征点匹配环节通过改进的 RANSAC 方法消除掉错误匹配,结果具有良好的鲁棒性与实时性。席志红等^[9]采用

PROSAC算法^[10]来剔除ORB-SLAM2系统中的误匹配,提高特征匹配度的精确性。刘畅等^[11]提出一种基于ORB-SLAM3的改进型特征匹配算法,提出基于一种四叉树的特征点提取方法,特征匹配环节引入GMS算法,降低匹配时间。季莘翔等^[12]采用同时在初始阈值和最小阈值中加入中间阈值,减少冗余特征点提取,提高图片特征点均匀性。潘峰等^[13]在ORB-SLAM算法的基础上提出分区域进行特征点提取并融合GMS算法剔除误匹配的改进型算法。马哲伟等^[14]提出基于自适应阈值的FAST特征点提取,再通过改进的四叉树均匀化策略对图像的特征点进行剔除与补偿,完成特征点选取,有效的提高了特征匹配的对数。

由于ORB-SLAM2算法特征点提取分布不均匀,在特征匹配时易出现误匹配,以及无法构建稠密点云地图,本文在特征匹配环节引入GMS算法^[15],利用基于网格剔除误匹配的统计量筛选匹配对,从而得到分布均匀的匹配对,并且根据关键帧与位姿构建稠密点云地图。

1 算法结构框架

ORB-SLAM2算法是一种基于特征点法的视觉SLAM算法,但缺少占据信息,对地图中的障碍物描绘模糊,只能构建稀疏地图^[8],因此,需要对ORB-SLAM2算法进行增强,本文算法整体框架如图1所示,其中虚线部分为本文改进的部分。

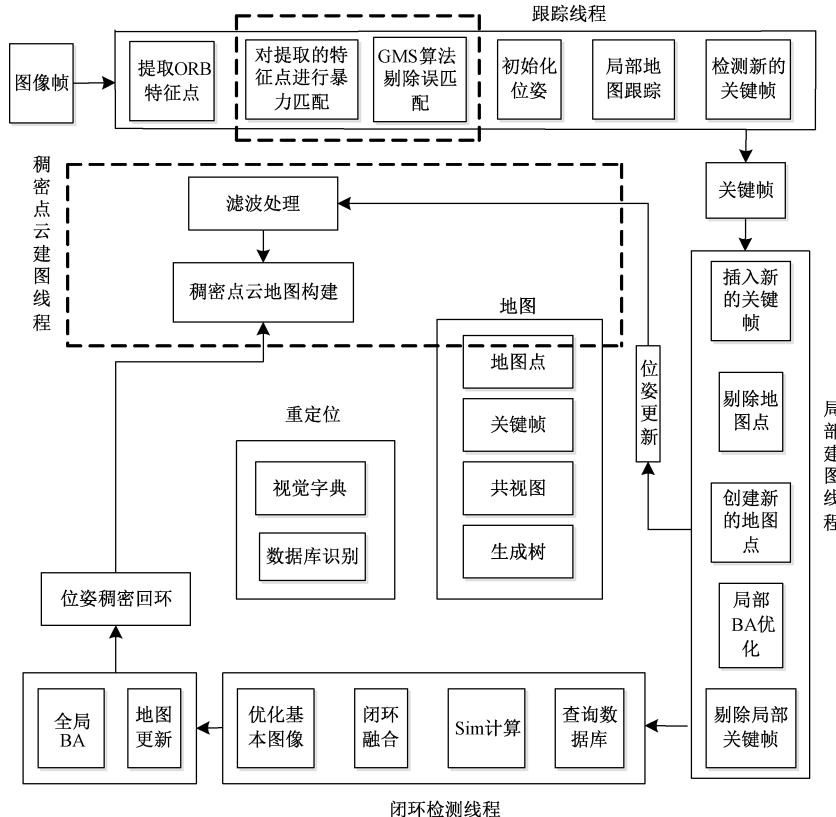


图1 系统框架图

Fig. 1 System framework diagram

算法主要分为以下几个部分:

1) 跟踪线程

该线程首先对图像进行预处理,从每个图像提取ORB特征点,根据前一帧的图像对当前帧进行优化,估计当前的位姿,同时通过相邻的地图点匹配更多的特征来进行优化,最后选取关键帧为建立局部地图作准备。由于原算法特征点分布不均匀导致误匹配率过高,本算法在特征匹配环节引入GMS匹配方法,从而得到分布较均匀且正确的匹配对。

2) 局部地图线程

将跟踪线程的关键帧插入地图中,筛选稀疏点云地图

点,剔除冗余关键帧和地图点,然后通过局部BA进一步优化位姿和地图点。

3) 稠密点云建图线程

提取稀疏点云地图点,根据关键帧、位姿估计与RGB-D的深度信息进行稠密地图的构建。

4) 闭环检测线程

该线程先使用词袋模型在关键帧数据库中确认候选闭环帧,采用Sim3变换计算当前帧与候选闭环帧的相似变换,然后校正关键帧位姿和局部地图点的三维坐标,融合地图点和位姿图优化^[16]。

2 ORB-SLAM2 特征提取

2.1 ORB 特征点

ORB 特征点由 FAST 关键点^[17]和 BRIEF 描述子^[18]组成。FAST 关键点是一种角点, FAST 关键点检测是一种快速检测图像局部特征的方法, 它主要通过比较像素与其邻域的灰度差异来识别角点, 以速度快著称。它的检测过程为: 在图像中选取一个像素点作为起始点, 设定一个阈值; 以这个像素点为中心, 画一个半径为 3 像素的圆, 此时圆上有 16 个像素点; 如果圆上有 n 个连续像素点, 并且这些像素点的灰度值比中心像素的灰度值大的多或者比中心像素的灰度值小的多, 那么这个中心像素点就认为是角点。BRIEF 描述子是一种二进制描述子, 描述量由多个 0 和 1 组成。

2.2 灰度质心法

FAST 角点不具有方向, 如果图像发生旋转, BRIEF 描述子也会随之发生变化, 导致特征点缺失, 所以通过灰度质心法的计算, 改进后的 FAST 被称为 Oriented FAST, 具有旋转和尺度的描述。

灰度质心法在图像区域内, 以一像素灰度值作为权重的中心, 形心指向质心的向量, 即为关键点的主方向。取某个角点一定半径内的像素灰度进行计算, 则对矩的定义为:

$$m_{pq} = \sum_{(x,y) \in r} x^p y^q I(x,y) \quad (1)$$

式中: $I(x,y)$ 为像素在坐标 (x,y) 处图像的灰度值; m_{pq} 为图像的矩在半径为 r 的图形图像区域; p,q 取值为 0 或 1。

则矩的质心为:

$$C = \left[\frac{m_{10}, m_{01}}{m_{00}, m_{00}} \right] \quad (2)$$

由上式的质心坐标可以计算出特征点的方向, 关键点的“主方向”可以表示为从圆形图像形心 O 指向质心 C 的方向, 则关键点的旋转角度为:

$$\theta = \arctan2(m_{01}/m_{10}) \quad (3)$$

2.3 四叉树策略筛选特征点

对于 ORB 算法提取的特征点容易出现扎堆以及过于集中的问题, 降低了特征点提取的效率, 因此对图片划分区域, 具体步骤如下:

1) 对图像金字塔上的每层都进行网格划分;

2) 通过将每层图像的 FAST 角点均匀分配到每个网格, 根据实际提取的数目调整阈值, 使算法在纹理区域较弱的区域都能检测到足够多的 FAST 角点;

3) 对每一个网格中的特征点采用四叉树原理, 若其中的一些网格里没有特征点, 则删除; 若其中的一些网格中有特征点, 则继续将有特征点的网格分成四份, 判断这四个网格中是否有特征点, 最后按照策略筛选出均匀分布的

FAST 角点。

四叉树策略筛选特征点具体过程如下:

(1) 先确定初始节点数目, 根据图像的宽度和高度将分辨率 640×480 的图像分成一个节点。一个图像初始化阶段只有一个节点, 这个节点就是四叉树的根节点。

(2) 第 1 次分裂后成 4 个节点, 有些特征点会被多个节点利用, 如图像中心的点就同时被 4 个节点使用, 统计 4 个节点的特征点数。若某个节点的特征点数为 0, 就删除该节点; 若特征点数为 1, 则不再分裂, 保留到分裂终止再处理; 若特征点数大于 1, 则该节点还可以分裂。判断此时的节点总数是否超过设点值, 若没有超过则继续进行分裂。

(3) 对于特征点数大于 1 的节点继续分裂, 一直重复过程(2)中的操作, 直到各节点不再分裂或节点数大于所需的特征点数时停止。

(4) 当分裂到达节点阈值时, 用非极大值抑制将角点响应值最大的特征点设为这个节点的代表特征点, 其他的特征点删除, 这样就得到了分布均匀的特征点。

四叉树策略筛选特征点如图 2 所示。

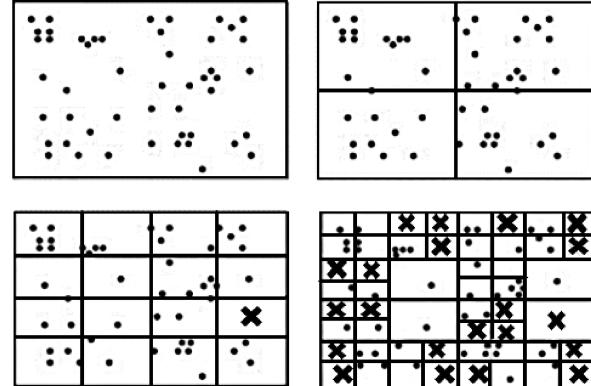


图 2 四叉树筛选特征点流程图

Fig. 2 Quadtree feature point filtering flowchart

3 改进型 ORB-SLAM2 算法

3.1 基于 GMS 算法的误匹配剔除

ORB-SLAM2 算法对特征点进行暴力匹配后会存在误匹配对, 本文通过在特征匹配环节引入 GMS 匹配算法剔除误匹配。该算法通过将运动平滑约束转换为剔除错误匹配的统计量, 对于正确匹配的邻域内匹配的数量多的概率很大, 而错误匹配的邻域内匹配的数量多的概率。

假设有两张图像 $\{I_a, I_b\}$, 分别有 N, M 个特征匹配, 定义 $C = \{c_1, c_2, c_3, \dots, c_i\}$ 表示图 I_a 到图 I_b 的最近邻匹配, 则:

$$N_i = \{c_j \mid c_j \in C, c_j \neq c_i, d(p_i, p_j) < r_1\} \quad (4)$$

$$S_i = \{c_j \mid c_j \in N_i, d(q_i, q_j) < r_2\} \quad (5)$$

式中: c_i 为像素点 p_i 和 q_i 的匹配; N_i 为 c_i 的邻域; S_i 为 c_i 的相似邻域; d 为两点的欧式距离; r_1, r_2 为距离阈值。

由二项分布估计可知:

$$|S_i| \sim \begin{cases} B(|N_i|, t), c_i \text{ is true} \\ B(|N_i|, \epsilon), c_i \text{ is false} \end{cases} \quad (6)$$

式中: $|S_i|$ 为 S_i 中元素数量, 对 c_i 的运动支持; $|N_i|$ 为 c_i 的邻域数量; S_i 为 c_i 的相似邻域; t 为正确匹配被某个邻域窗口匹配支持的概率; ϵ 为错误匹配被某个邻域窗口匹配支持的概率。

通过计算分析可得:

$$E_{|S_i|} = \begin{cases} E_t = |N_i| \cdot t, c_i \text{ is true} \\ E_f = |N_i| \cdot \epsilon, c_i \text{ is false} \end{cases} \quad (7)$$

$$V_{|S_i|} = \begin{cases} V_t = |N_i| \cdot t \cdot (1-t), c_i \text{ is true} \\ V_f = |N_i| \cdot \epsilon \cdot (1-\epsilon), c_i \text{ is false} \end{cases} \quad (8)$$

式中: $E_{|S_i|}$ 为 $|S_i|$ 的期望值; $V_{|S_i|}$ 为 $|S_i|$ 的方差。

通过设置阈值, 用期望值和方差可表示为:

$$P = \frac{|E_t - E_f|}{\sqrt{V_t} + \sqrt{V_f}} = \frac{|N_i| \cdot (t - \epsilon)}{\sqrt{|N_i|} \cdot t \cdot (1-t) + \sqrt{|N_i|} \cdot \epsilon \cdot (1-\epsilon)} \quad (9)$$

式中: P 为正确和错误匹配的可区分度, $P \propto \sqrt{|N_i|}$ 。

利用划分网格的思想可以对以上算法进行加速, 将图像划分成多个小网格, c_i 表示落在网格 G_a 和 G_b 中的一个匹配对, 如图 3 所示, 图 3 中用实线段表示。那么 c_i 的邻域和相似邻域被重新定义为:

$$N_i = \{c_j \mid c_j \in C_a, c_i \neq c_j\} \quad (10)$$

$$S_i = \{c_j \mid c_j \in C_{ab}, c_i \neq c_j\} \quad (11)$$

式中: C_a 为网格 G_a 中的匹配对; C_{ab} 为同时在网格 G_a 和网格 G_b 中的匹配对。

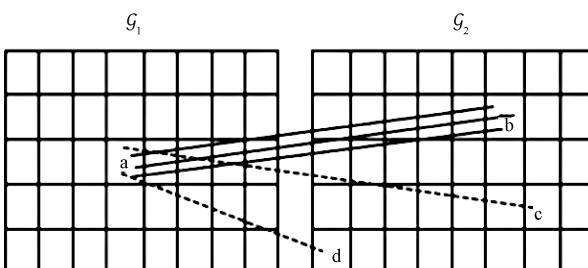


图 3 基于网格的匹配

Fig. 3 Grid based matching

若网格很小, 则很多邻域的信息容易忽略, 若网格过大, 则会将不正确的匹配关系包含。因此采用运动核的方式解决, 如图 4 所示, 设置 3×3 的网格, 将原来邻域内匹配点问题转为网格对应匹配问题。

定义网格的邻域为周围的 8 个网格 ($C_{a^1 b^1}, C_{a^2 b^2}, \dots, C_{a^9 b^9}$), 因此 c_i 的邻域和相似邻域被重新定义为:

$$N_i = \{c_j \mid c_j \in C_A, c_i \neq c_j\} \quad (12)$$

$$S_i = \{c_j \mid c_j \in C_{AB}, c_i \neq c_j\} \quad (13)$$

式中: C_A 为网格中 G_a 的匹配对; C_{AB} 为网格 A 和网格 B

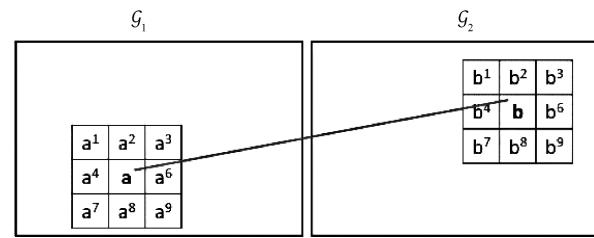


图 4 运动核

Fig. 4 Sports core

中的匹配对, $C_{AB} = C_{a^1 b^1} \cup C_{a^2 b^2} \cup \dots \cup C_{a^9 b^9}$ 。

为了解决在匹配过程中出现尺度和旋转问题, 该算法采用多尺度多旋转的策略。针对尺度问题, 具体为固定一个网格图像, 大小为 $n \times n$, 则另一个网格图像大小为 $(n \cdot \alpha) \times (n \cdot \alpha)$, 则 α 有 5 种取值方式 $\left\{\frac{1}{2}, \frac{\sqrt{2}}{2}, 1, \sqrt{2}, 2\right\}$ 。针对旋转问题, 利用旋转运动核不同的旋转方向, 固定 G_a , 对 G_b 顺势针旋转后可以得到 8 个运动核, 如图 5 所示。

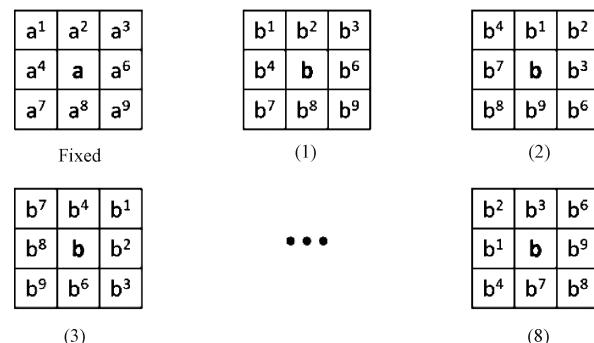


图 5 旋转运动核

Fig. 5 Rotational motion nucleus

本文引入 GMS 算法的改进型算法流程如图 6 所示。

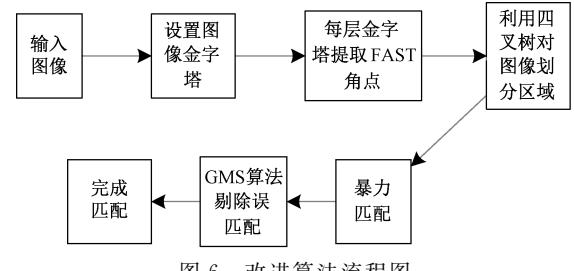


图 6 改进算法流程图

Fig. 6 Improve algorithm flowchart

3.2 基于稠密点云地图的构建

对于定位而言, ORB-SLAM2 构建的稀疏特征地图就足够使用, 而要用于导航和路径规划, 需要稠密点云地图。因此对 ORB-SLAM2 算法进行改进, 使其能构建稠密点云地图。本文在前面改进算法的基础上添加稠密点云地图的构建, 流程如图 7 所示。

稠密点云地图的构建主要包括 3 个阶段, 输入数据、位姿估计、建图。首先输入 RGB 图像、深度图和特征点;

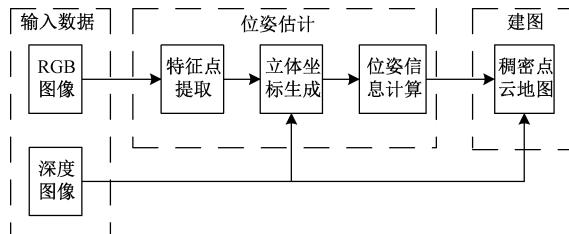


图 7 稠密点云建图结构图

Fig. 7 Dense point cloud mapping structure diagram

其次对关键帧进行位姿估计,构建局部稠密点云地图;最后,接收图像数据和位姿信息,对局部点云地图进行拼接,得到全局稠密点云地图。

生成点云地图,其原理是相机针孔成像,它的数学模型为:

$$a \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \mathbf{W} \cdot (\mathbf{R} \cdot \mathbf{T}_i + \mathbf{Q}) = \begin{pmatrix} f_x & 0 & q_x \\ 0 & f_y & q_y \\ 0 & 0 & 1 \end{pmatrix} \left(\mathbf{R} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} + \mathbf{Q} \right) \quad (14)$$

式中: u 为像素的平面坐标横轴; v 为像素的平面坐标纵轴; \mathbf{W} 为相机内部参数矩阵; \mathbf{R} 为旋转矩阵; \mathbf{T}_i 为空间点坐标; \mathbf{Q} 为平移矩阵; a 为深度值与实际空间距离比例关系因子。

完成相机标定后 \mathbf{W} 为已知量,追踪线程进行位姿估计后 \mathbf{R} 和 \mathbf{Q} 确定,利用得到的图像深度信息来获取点云数据:

$$\begin{bmatrix} p_x \\ p_y \\ p_z \\ h \end{bmatrix} = \begin{bmatrix} (n - q_x) \cdot p_z / f_x \\ (n - q_y) \cdot p_z / f_y \\ h \end{bmatrix} \quad (15)$$

式中: p 一点云数据点; q_x 为相机光轴在图像平面坐标 x 轴的偏移量; q_y 为相机光轴在图像平面坐标 y 轴的偏移量; f_x 为 x 轴上的相机焦距; f_y 为 y 轴上的相机焦距; h 为深度图像中点的深度值。

得到点云数据后,生成局部点云地图,与生成的地图进行点云拼接以及全局优化,最后得到稠密点云地图。

4 仿真实验与分析

实验运行环境:操作系统为Vmware虚拟机Ubuntu20.04LTS,CPU为Intel(R)Core(TM)i7-8750H,主频2.20GHz,内存为8G,开发语言为C++。

4.1 改进算法特征点提取与匹配仿真实验

本文采用两帧相同的图像进行特征点提取的对比实验,图8为ORB算法与ORB-SLAM2算法特征提取的效果图。

如图8(a)所示,传统ORB算法提取的特征点绝大多数在集中在一个区域,甚至一部分区域没有提取特征点,这样会导致图片信息没有利用,而图8(b)在引入四叉树策略的情况下,提取的特征点分布均匀,在图像细节较弱的



图 8 不同算法特征点提取效果

Fig. 8 The effectiveness of feature point extraction using different algorithms

区域,仍然能有效的提取到特征点,从而确保图像帧被最大提取利用。

本文采用两帧相近的图像进行特征匹配的对比实验,为提高实验真实性和准确性,从TUM数据集的fr1_xyz序列里取两帧相近的图像,将本文算法分别与暴力匹配算法、ORB-SLAM2算法进行对比,结果如图9所示,效果对比如表1所示。

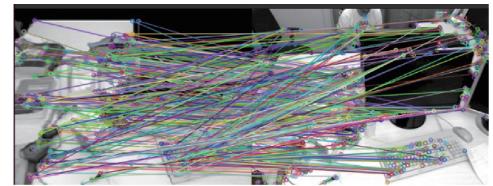
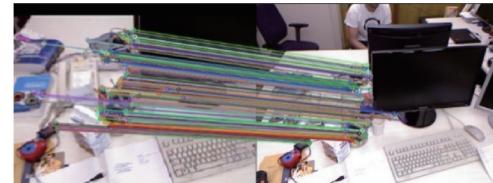
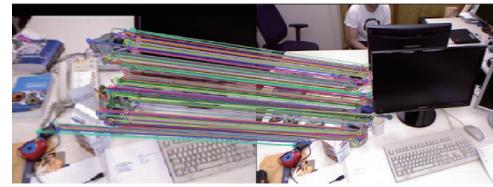
(a) 暴力匹配结果
(a) Brute-force matching results(b) ORB-SLAM2算法结果
(b) Results of the ORB-SLAM2 algorithm(c) 本文算法结果
(c) Results of the proposed algorithm

图 9 不同算法特征匹配效果图

Fig. 9 Feature matching of different algorithms

表 1 不同算法特征匹配效果对比

Table 1 Comparison of feature matching effects of different algorithms

值	暴力匹配	ORB-SLAM2	改进后算法
提取个数	834	989	1 000
匹配对数	371	294	453
匹配用时/s	0.135 6	0.008 2	0.007 5
匹配率/%	44.52	29.75	45.3

由表 1 分析可知, 改进算法的匹配数比原本 ORB-SLAM2 算法增加了 18.36%, 匹配用时比与原本算法减少了 8.53%, 并且从图 9 可知, 本文算法相较于暴力匹配能得到较均匀的匹配对, 明显降低了匹配成本, 从而验证了本文算法的优越性。

如表 2 所示, 通过对 TUM 数据集中的 fr1_xyz、fr1_room、fr1_desk 三组序列上相近两帧图像的匹配率进行分析, 改进型 ORB-SLAM2 算法的匹配率总体上比原来 ORB-SLAM2 算法增加了 52.5%, 由此得出, 改进型 ORB-SLAM2 算法更加优越。

表 2 在三组数据集上的匹配率

Table 2 Match rate on the three datasets %

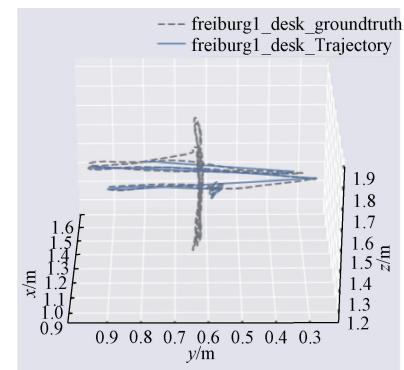
值	暴力匹配	ORB-SLAM2	改进型算法
fr1_xyz	44.52	29.75	45.30
fr1_room	33.84	25.62	41.91
fr1_desk	29.49	20.87	35.95
平均	35.95	25.41	41.07

4.2 改进算法精度仿真实验

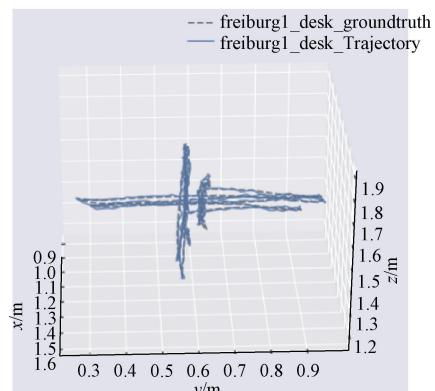
对于改进型 ORB-SLAM2 算法的定位准确度, 使用 evo 评估工具对其进行测评, 测评结果如图 10 所示, 图 10 为 ORB-SLAM2 算法与改进后算法在 TUM 数据集的 fr1_xyz 序列上的真实轨迹与预测轨迹图, 其中 freiburg1_desk_groundtruth 表示以预测轨迹, freiburg1_desk_trajectory 表示真实轨迹。根据对比两条轨迹线段的偏差, 分析得到本文改进后的算法偏差相比原来算法更小且更接近真实轨迹。

本文采用绝对轨迹误差(APE)对定位精度进行对比, 绝对轨迹误差越小, 定位精度越高, 对比结果如图 11 所示, 图 11 为 ORB-SLAM2 算法与改进后的算法在 TUM 数据集的 fr1_xyz 序列上的绝对轨迹误差图, 表 3 为 3 种算法绝对轨迹误差统计指标, 文献[19]中主要针对 ORB-SLAM2 算法进行改进。经过对比分析, 本文改进型算法的各指标也相对更小, 相较于 ORB-SLAM2 算法精度提高了 3.62%, 鲁棒性更强。

为进一步验证改进型 ORB-SLAM2 算法的效果, 本文进行了轨迹测试实验, 如图 12 所示, 分别为 ORB-SLAM2



(a) ORB-SLAM2 算法轨迹
(a) Trajectory of the ORB-SLAM2 algorithm



(b) 改进型算法轨迹
(b) Improved algorithm trajectory

图 10 真实迹与预测轨迹对比

Fig. 10 Comparison between the true trajectory and the predicted trajectory

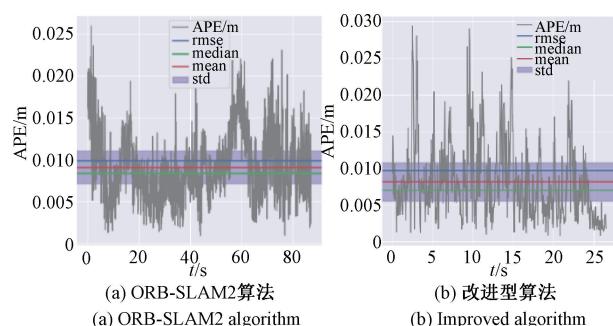


图 11 绝对轨迹误差曲线

Fig. 11 Absolute trajectory error curve

表 3 绝对轨迹误差统计指标对比

Table 3 Comparison of absolute trajectory error statistical indicators

算法类型	最大值/m	平均误差/m	误差中值/m	最小值/m	均方根误差/m	方差/m ²	标准差/m
ORB-SLAM2	0.0259	0.0113	0.0101	0.0004	0.0130	0.0049	0.0063
文献[19]算法	0.0332	0.0082	0.0071	0.0002	0.0097	0.0750	0.0052
改进型算法	0.0251	0.0045	0.0038	0.0001	0.0054	0.0236	0.0030

算法与改进型 ORB-SLAM2 算法在 xyz 坐标轴的 rpy 角的轨迹曲线, 其中 freiburg1_desk_trajectory 表示 ORB-

SLAM2 算法, freiburg1_desk_trajectory 表示改进型 ORB-SLAM2 算法。

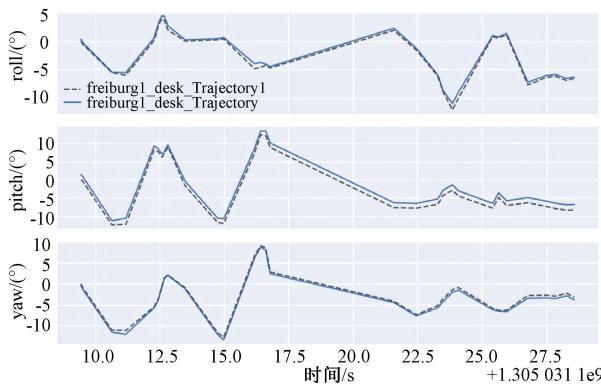


图 12 xyz 坐标轴的 rpy 角的轨迹曲线

Fig. 12 Trajectory curve of rpy angle on xyz coordinate axis

结果表明 ORB-SLAM2 算法轨迹长度为 2.152 m, 改进型 ORB-SLAM2 算法轨迹长度为 2.431 m, 轨迹增加了 12.9%, 改进型 ORB-SLAM2 算法在特征点较少时还可以继续跟踪, 由此可见, 改进型 ORB-SLAM2 算法在总体上提高了定位精度。

4.3 改进算法稠密点云地图仿真实验

为了验证改进型算法的性能效果, 使用 RGB-D 相机获得的深度信息生成三维空间点云信息, 本文在 TUM 数据集的 fr1_xyz、fr1_room、fr1_desk 三组序列上进行稠密点云地图实验, 建图效果如图 13 所示, 可以看出桌子上的电脑、地面和人物等都可以清楚的展现出来, 因此分析说明本文构建的稠密点云地图效果较好, 可以用于移动机器人进一步导航等用途。

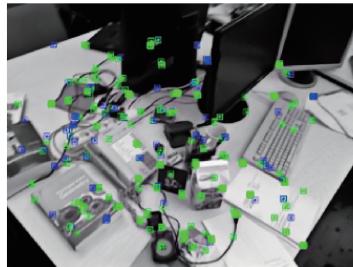
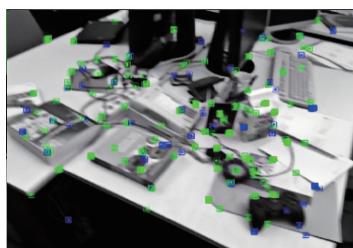
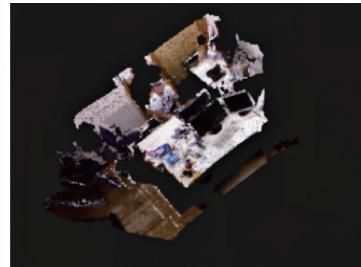
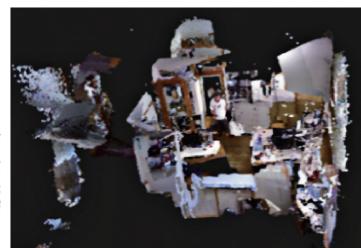
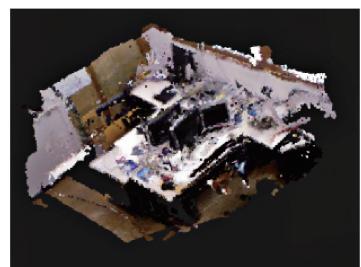
(a) fr1_xyz 稠密点云地图
(a) fr1_xyz dense point cloud map(b) fr1_room 稠密点云地图
(b) fr1_room dense point cloud map(c) fr1_desk 稠密点云地图
(c) fr1_desk dense point cloud map

图 13 改进型算法在三组数据集下的稠密点云地图构建

Fig. 13 Improved algorithm for constructing dense point cloud maps on three datasets

5 结论

本文提出了一种基于 ORB-SLAM2 的改进型特征匹配算法, 并在这基础上增加稠密点云地图线程, 主要工作

如下: 建立图像金字塔, 对金字塔上的图像进行网格划分, 提取特征点; 利用四叉树思想, 对每个网格中的特征点进行筛选, 得到均匀分布的特征点; 特征匹配阶段引入 GMS 算法剔除误匹配, 相较于原来的 ORB-SLAM2 算法, 匹配

率明显提高,匹配数也增加,匹配耗时也明显低于原始算法;在改进算法基础上添加稠密点云地图线程,建立的地图清晰的显示了环境信息。本文在 TUM 数据集上进行实验验证,结果表明本文算法在总体上提高了定位精度。针对 GMS 算法需要提取较多的特征点以提高正确匹配度的问题,本文下一步将优化特征点提取方法并将稠密点云地图转化为八叉树地图应用于机器人导航。

参考文献

- [1] 吴玉香,王超,洗颖宪. 基于稀疏直接法和图优化的移动机器人 SLAM[J]. 仪器仪表学报, 2018, 39(4): 257-263.
WU Y X, WANG CH, XIAN Y X. SLAM based on sparse direct method and graph optimization for mobile robot[J]. Chinese Journal of Scientific Instrument, 2018, 39(4): 257-263.
- [2] JIANG W, YE G CH, ZHOU D H. Dynamic model based energy consumption optimal motion planning for high-voltage transmission line mobile robot manipulator[J]. Institution of Mechanical Engineers, Part K: Journal of Multi-body Dynamics, 2021, 253 (1): 93-105.
- [3] 何俊学,李战明. 基于视觉的同时定位与地图构建方法综述[J]. 计算机应用研究, 2010, 27 (8): 2839-2844.
HE J M, LI ZH M. Survey of vision-based approach to simultaneous localization and mapping[J]. Application Research of Computers, 2010, 27(8): 2839-2844.
- [4] 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 SH, GUO Y X, GAO T X, et al. RGB-D visual SLAM algorithm for mobile robots[J]. Transactions of the Chinese Society for Agricultural Machinery, 2018, 49(10):38-45.
- [5] KAZEROUNI I A, FITZGERALD L, DOOLY G, et al. A survey of state-of-the-art on visual SLAM[J]. Expert Systems with Applications, 2022, 205(11):117734.
- [6] MUR-ARTAL R, TARDOS J D. ORB-SLAM2: An open-source slam system for monocular, stereo, and RGBD cameras[J]. IEEE Transactions on Robotics, 2017, 33(5): 1255-1262.
- [7] JIANG W, YE G C, ZHOU D H. Dynamic model based energy consumption optimal motion planning for high-voltage transmission line mobile robot manipulator[J]. Institution of Mechanical Engineers, Part K: Journal of Multi-body Dynamics, 2021, 253(1): 93-105.
- [8] 谷学静,郭俊,刘海望,等. 基于四叉树分解法改进的 ORB 特征匹配算法[J]. 激光杂志, 2021, 42(6):56-61.
GU X J, GUO J, LIU H W, et al. An improved ORB feature matching algorithm based on quadtree decomposition method [J]. Laser Journal, 2021, 42(6):56-61.
- [9] 席志红,王洪旭,韩双全. 基于 ORB-SLAM2 系统的快速误匹配剔除算法与地图构建[J]. 计算机应用, 2020, 40(11): 3289-3294.
XI ZH H, WANG H X, HAN SH Q. Fast mismatch elimination algorithm and map-building based on ORB-SLAM2 system [J]. Journal of Computer Applications, 2020, 40(11): 3289-3294.
- [10] CHUM O, MATAS J. Matching with PROSAC—progressive sample consensus [C]. 2005 IEEE Conference on Computer Vision and Pattern Recognition. Piscataway:IEEE, 2005:220-226.
- [11] 刘畅,党淑雯,陈丽. 基于 ORB-SLAM3 的改进型特征匹配与稠密建图算法[J]. 计算机应用研究, 2023, 40(11): 3443-3449.
LIU CH, DANG SH W, CHEN L. Improved feature matching and dense-mapping algorithm based on ORB-SLAM3[J]. 2023, 40(11): 3443-3449.
- [12] 季莘翔,王宇钢,林一鸣. 改进 ORB 特征提取算法的 ORB-SLAM2 定位研究[J]. 组合机床与自动化加工技术, 2023(11): 60-64.
JI SH X, WANG Y G, LIN Y M. Research on ORB-SLAM2 localization based on improved ORB feature extraction algorithm[J]. Modular Machine Tool & Automatic Manufacturing Technique, 2023 (11): 60-64.
- [13] 潘峰,沈建新,秦顺,等. 融合 GMS 的 ORB 特征点提取与匹配算法[J]. 计算机工程与设计, 2022, 43(8): 2244-2251.
PAN F, SHEN J X, QIN SH, et al. Improved algorithm of ORB feature point extraction and matching based on GMS[J]. Computer Engineering and Design, 2022,43(8):2244-2251.
- [14] 马哲伟,周福强,王少红. 昏暗环境下自适应 ORB-SLAM2 算法研究[J]. 电子测量技术, 2024, 47 (6): 94-99.
MA ZH W, ZHOU F Q, WANG SH H. Research on adaptive ORB-SLAM2 algorithm in dark environments [J]. Electronic Measurement Technology, 2024, 47(6):

- 94-99.
- [15] BIAN J W, LIN W Y, MATSUSHITA Y, et al. GMS: Grid-based motion statistics for fast, ultra-robust feature correspondence [C]. IEEE CVPR, 2017: 4181-4190.
- [16] 田野, 陈宏巍, 王法胜, 等. 室内移动机器人的 SLAM 算法综述[J]. 计算机科学, 2021, 48(9): 223-234.
TIAN Y, CHEN H W, WANG F SH, et al. Overview of SLAM algorithms for mobile robots[J]. Computer Science, 2021, 48(9): 223-234.
- [17] ROSTEEN E, DRUMMOND T. Machine learning for high-speed corner detection [C]. Computer Vision-ECCV, 2006: 430-443.
- [18] CALONDER M, LEPEYIYV, STRECHA C, et al. Brief: Binary robust independent elementary features [C]. Computer Vision-ECCV, 2010: 778-792.
- [19] 刘庆运, 杨华阳, 刘涛, 等. 基于激光雷达与深度相机融合的 SLAM 算法[J]. 农业机械学报, 2023, 54(11): 29-38.
LIU Q Y, YANG H Y, LIU T, et al. SLAM algorithm based on fusion of Lidar and depth camera [J]. Transactions of the Chinese Society for Agricultural Machinery, 2023, 54(11): 29-38.

作者简介

王喜红, 硕士研究生, 主要研究方向为机器人多传感器融合 slam。

E-mail: 1508473660@qq.com

雷斌(通信作者), 博士, 副教授, 主要研究方向为多传感器融合 slam、深度学习。

E-mail: leibin@mail.lzjtu.cn