

# 基于改进关键帧选择的 RGB-D SLAM 算法

李弋星, 刘士荣\*, 仲朝亮, 王 坚

(杭州电子科技大学 自动化学院, 浙江 杭州 310018)

**摘要:** 关键帧选择是提高视觉 SLAM(simultaneous localization and mapping)算法精度及实时性的重要因素. 关键帧常以图像的帧间相对运动距离为选择依据. 该方法虽简单有效, 但实时性、鲁棒性较差且容易产生大量冗余关键帧. 针对上述问题, 提出一种改进的关键帧选择算法. 该算法整合了帧间相对运动距离、帧间特征点跟踪以及最小视觉变化来选择关键帧并删除冗余关键帧. 基于该算法, 结合具有较好方向和光照不变性的 ORB(oriented FAST and rotated BRIEF)特征, 实现了 RGB-D SLAM 算法. 在 RGB-D 数据集上的实验表明, 改进的关键帧选择算法能够更精准、及时地选择关键帧, 并在减少 RGB-D SLAM 中冗余关键帧的同时提高算法的实时性、建图和定位精度.

**关键词:** 关键帧选择; 冗余关键帧删除; ORB 特征; RGB-D SLAM

**中图分类号:** TP242

**文献标识码:** A

**doi:** 10.7511/dllgxb201704012

## 0 引言

近年来, 机器人三维同时定位与建图(3D SLAM)问题已经成为机器人领域研究的热点<sup>[1]</sup>. 相较于二维 SLAM, 三维 SLAM 能够更好地利用环境的三维属性, 描述出环境中物体的空间几何外形, 对机器人在未知环境中的导航具有重要的意义<sup>[2]</sup>. 得益于视觉传感器信号探测范围宽、信号获取内容全、价格低廉等特点, 国内外学者对 VSLAM<sup>[3]</sup>(visual SLAM)进行了大量研究, 包括 LSD SLAM<sup>[4]</sup>、ORB-SLAM<sup>[5]</sup>、RGB-D SLAM<sup>[6]</sup>等.

TUM(Technische Universität München)计算机视觉组提出的 LSD SLAM 直接利用单目相机实现三维 SLAM. 不同于 ORB-SLAM 采用特征点进行帧间匹配, LSD SLAM 采用直接法, 根据灰度图强度(image intensities)进行帧间匹配和建图, 实现了大规模环境下的准确定位与建图. ORB-SLAM 利用 ORB 特征完成 SLAM 中的所有任务: 跟踪、建图、重定位和环形闭合检测. 在室内、室外以及大规模环境下, ORB-SLAM 有较好的鲁棒性和实时性. 与 LSD SLAM 使用单目相

机通过三角测量方法恢复环境深度相比, RGB-D 相机(如微软公司的 Kinect)能够在捕获 RGB 图像的同时获得每一个像素的深度信息, 很大程度上简化了利用三角测量方法获取深度信息的过程. 基于上述优点, 德国弗莱堡大学 Endres 等<sup>[6]</sup>提出了 RGB-D SLAM 算法, 使用视觉传感器 Kinect 作为唯一传感器对周围环境进行三维建图. 实验证明, RGB-D SLAM 算法能快速建图并具有很好的鲁棒性.

在 VSLAM 中, 通常使用关键帧进行建图和定位, 这样可以避免因逐帧插入导致系统实时性降低、计算代价增加、系统内存消耗过大的问题. 迄今为止, 根据帧间相对运动距离来选择关键帧是 VSLAM 最常用的方法<sup>[7]</sup>. 该方法虽简单有效, 但精度低、实时性差. 同时该方法往往直接将已选择的关键帧全部储存<sup>[1,6]</sup>, 导致 VSLAM 算法后期存在大量冗余关键帧. 本文针对上述问题, 首先提出一种结合帧间相对运动距离、特征点跟踪以及最小视觉变化的关键帧选择算法, 以便更加及时、准确地插入关键帧. 然后通过关键帧之间的图像可跟踪点进行二次判断, 删除冗余关键帧. 最后基于改

收稿日期: 2016-12-30; 修回日期: 2017-05-26.

基金项目: 国家自然科学基金资助项目(61175093).

作者简介: 李弋星(1992-), 女, 硕士生, E-mail: yixing\_li\_323@163.com; 刘士荣\*(1952-), 男, 教授, 博士生导师, E-mail: liushirong@hdu.edu.cn.

进的关键帧选择算法,结合具有较好方向和光照不变性的 ORB 特征,实现 RGB-D SLAM 算法。

## 1 RGB-D SLAM 算法框架

本文通过基于图(graph-based)的方法实现 RGB-D SLAM 算法.该算法主要由前端(front-end)、后端(back-end)、地图构建 3 部分组成,如图 1 所示.前端又叫位姿图构建;后端又叫图优化,用以优化全局位姿图。

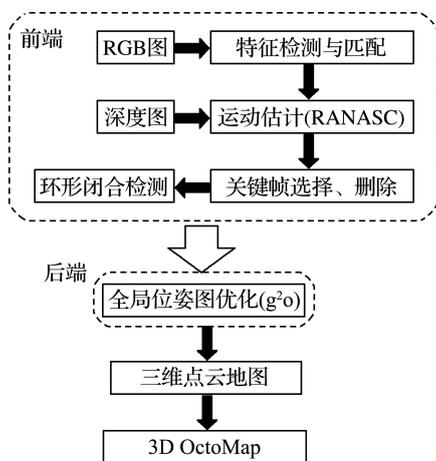


图 1 RGB-D SLAM 的算法框架

Fig. 1 The overview of RGB-D SLAM algorithm

### 1.1 图构建

图构建主要包括特征检测与匹配、运动估计、关键帧选择与冗余关键帧删除以及环形闭合检测 4 部分。

常用的特征点包括 SIFT<sup>[8]</sup>、SURF<sup>[9]</sup>、FAST<sup>[10]</sup>等.考虑到 SLAM 对实时性的要求,采用 ORB<sup>[11]</sup>特征进行特征检测与提取.ORB 特征由 Rublee 等<sup>[11]</sup>于 2011 年提出,其计算速率更快、精度更高且具有更好的鲁棒性(尺度和旋转不变性).对于 ORB 特征,采用 BruteForceMatcher 结合汉明距离进行匹配.由于图像匹配后会包含大量误匹配对,采用随机采样一致性算法(RANSAC)<sup>[12]</sup>并利用 RGB-D 相机提供的深度信息对初匹配结果进行采样,从而剔除外点(outliers)、抑制噪声并利用正确的匹配对计算帧间匹配的内点(inliers)、旋转向量和平移向量;然后根据旋转向量和平移向量进行运动估计,计算出帧间相对运动距离;再结合特征点跟踪和最小视觉变化进行关键帧选择,之后通过关键帧之间的图像可跟踪点进行二次判断,删除冗余关键帧;最后,再次利

用 RANSAC 算法进行局部环形闭合检测和全局环形闭合检测,用以检索当前场景是否重复出现,从而完成 SLAM 算法前端的所有环节。

### 1.2 图优化

完成图构建后,通过优化位姿图可以求解相机的最优位姿.常用的图优化方法包括非线性最小二乘法<sup>[13]</sup>、基于松弛的优化方法<sup>[14]</sup>、基于随机梯度下降法<sup>[15]</sup>以及流形优化方法<sup>[16]</sup>.非线性最小二乘法、基于松弛的优化方法和基于随机梯度下降法都是在欧氏空间中进行优化,但机器人的真实位姿变化却是在非欧氏空间中的.为了防止欧氏空间中奇异值的出现,使用四元数来表示机器人的位姿变化并在流形空间中进行优化.这样不仅克服了欧氏空间中产生奇异值的问题,还可取得更高的精度.本文具体采用  $g^2o$  进行后端优化. $g^2o$  是 Kümmerle 等<sup>[16]</sup>提出能用于流形优化的开源工具,可以大大提高开发效率。

### 1.3 地图构建

三维环境地图有很多表示方法,本文采用 3D 点云地图和 3D OctoMap<sup>[17]</sup>来建图.3D 点云地图能够清晰直观地表示环境信息,但存在大量冗余点且系统内存占用较大,不适合在大规模环境以及终生 SLAM(life-long SLAM)算法中使用.3D OctoMap 克服了点云地图上述缺点,在表示地图的同时占用少量的内存.此外,3D OctoMap 能够更加直观地表示环境中的占用、空闲和未知区域,更有利于机器人导航和路径规划。

## 2 关键帧选择算法

### 2.1 常用关键帧选择算法

当前端完成图像匹配后,利用 RGB-D 相机提供的各像素深度信息和 RANSAC 算法,可计算出帧间旋转向量  $\mathbf{r}$  和平移向量  $\mathbf{t}$ ,从而求解帧间相对运动距离:

$$D = \|\Delta \mathbf{t}\| + \min(2\pi - \|\mathbf{r}\|, \|\mathbf{r}\|) \quad (1)$$

以  $D$  为判断依据,常用关键帧选择算法如下:

① 若  $D_{\min} \leq D \leq D_{\max}$ , 则  $Frame_{curr} = Frame_{key}$ ;

② 若  $D < D_{\min}$ , 则  $Frame_{curr} \neq Frame_{key}$ ;

③ 若  $D > D_{\max}$ , 则  $Frame_{curr} \neq Frame_{key}$

其中,当前帧为  $Frame_{curr}$ ,关键帧为  $Frame_{key}$ ,两帧之间的最小运动距离为  $D_{\min}$ ,两帧之间的最大运动距离为  $D_{\max}$ .条件②说明两帧之间运动距离较近,变化较小;条件③说明两帧之间运动距离较远,

变化较大, 以上两种情况均说明当前帧不是关键帧。

除此之外, 还可以结合 RANSAC 算法中 *Inliers* 来选择关键帧. 该算法虽然结合了  $D$  和 *Inliers* 来判断是否选择关键帧, 但仍不能很好地满足 SLAM 算法对实时性、鲁棒性和精度的要求. 在一些相机旋转或抖动的特殊情况下仍会出现判断错误甚至关键帧丢失的现象. 对于终生 SLAM 算法来说, 由于没有判定冗余关键帧, 还会导致系统存储大量冗余的关键帧, 使计算和存储负荷不断增大.

为了克服常用关键帧选择存在的不足, 本文提出选择关键帧后, 检测冗余关键帧, 并将其删除.

## 2.2 改进关键帧选择算法

### 2.2.1 关键帧的选择

对  $D$  加以限制, 能确保帧间运动距离在一定的范围内, 对 *Inliers* 加以限制, 能够确保相邻图像之间有足够的匹配精度. 而特征点跟踪能够保证帧间的相关性, 提高建图的一致性. 本文提出最小视觉变化概念. 最小视觉变化是指两帧图像之间存在足够小但又满足需求的运动变化, 包括运动距离最小变化和帧间旋转角度最小变化. 最小视觉变化能够提高帧间匹配精度, 保证帧间相关性及建图的一致性.

为此, 在常用关键帧选择算法中引入了特征点跟踪和最小视觉变化作为关键帧判断依据. 该算法伪代码描述如下:

If (环形闭合检测空闲) or (距  $Keyframe_L$  插入后又经过了  $z$  个帧) Then

If ( $Inliers > Inliers_{min}$ ) Then

If ( $D_{min} \leq D \leq D_{max}$ ) Then

If ( $Track(Frame_{curr-1}, Frame_{curr}) > \alpha$ ) Then

If ( $Match(Keyframe_L, Frame_{curr}) > \beta$ )

Then

选择关键帧

EndIf

EndIf

EndIf

EndIf

EndIf

在上述算法中,  $Inliers_{min}$  为最少内点数,  $Frame_{curr}$  为当前帧,  $Frame_{curr-1}$  为上一帧,  $Keyframe_L$  为最新插入的关键帧,  $\alpha$ 、 $\beta$ 、 $z$  为可调参数.  $Track(Frame_{curr-1}, Frame_{curr})$  用以计算  $Frame_{curr}$  与  $Frame_{curr-1}$  之间跟踪到的特征点个数,  $Match(Keyframe_L, Frame_{curr})$  用以计算  $Frame_{curr}$

与  $Keyframe_L$  之间匹配的特征点个数占  $Keyframe_L$  特征点个数的比例.

首先, 判定是否插入关键帧, 即在不进行环形闭合检测时, 插入关键帧能够保证环形闭合检测的准确率、有效性. 在距  $Keyframe_L$  插入后检测到第  $z$  个非关键帧时插入关键帧, 能够确保在建图空闲时插入, 提高关键帧的检测效率.

其次, 利用帧间相对运动距离  $D$  以及内点数 *Inliers* 对关键帧进行筛选. 例如, 算法语句 If ( $Track(Frame_{curr-1}, Frame_{curr}) > \alpha$ ) Then 用来对相邻帧间特征点进行跟踪. 条件  $Track(Frame_{curr-1}, Frame_{curr}) > \alpha$  说明帧间相关性强、匹配度高, 将  $Frame_{curr}$  作为候选关键帧, 从而避免关键帧丢失.

最后, 以最小视觉变化进行判定, 检测  $Frame_{curr}$  与  $Keyframe_L$  之间跟踪到的特征点个数. 当  $Match(Keyframe_L, Frame_{curr}) > \beta$  时, 说明新插入的关键帧与当前帧之间有足够运动距离, 帧间旋转角度满足要求且匹配度高, 将  $Frame_{curr}$  选定为关键帧, 并在位姿图中插入该帧.

### 2.2.2 冗余关键帧的删除

大部分关键帧算法都不会检测冗余关键帧, 但检测冗余关键帧并将其删除不仅能在很大程度上保证 SLAM 算法构建地图的一致性, 还能减少关键帧的数量和占用的系统存储空间. 这对于需要检测环境变化并对地图进行持续更新和维护的终生 SLAM 算法, 具有重大意义.

本文通过比较相邻  $n$  个关键帧图像间的可跟踪特征点进行关键帧的二次判断. 冗余关键帧具体检测方法如下:

①  $Track(Keyframe_{curr-1}, Keyframe_{curr}) > \epsilon$ ;

② 重复上述比较方法;

③  $Track(Keyframe_{curr-n}, Keyframe_{curr}) > \epsilon$ ;

④ 如果以上判别式均成立,  $Keyframe_{curr}$  为冗余关键帧, 将其删除.

上述检测方法中,  $Keyframe_{curr-1}, \dots, Keyframe_{curr-n}$  为  $Keyframe_{curr}$  之前的  $n$  个关键帧,  $Keyframe_{curr}$  为当前关键帧,  $Track(Keyframe_{curr-n}, Keyframe_{curr})$  用以计算  $Keyframe_{curr}$  与  $Keyframe_{curr-n}$  之间匹配的特征点个数.

当  $Keyframe_{curr}$  上占比  $\epsilon$  ( $\epsilon > 70\%$ ) 的特征点能够在  $Keyframe_{curr-n}$  上检测到时, 说明  $Keyframe_{curr}$  相对于之前的  $n$  个关键帧的运动尺

度过小,  $Keyframe_{curr}$  为冗余关键帧, 删除  $Keyframe_{curr}$ . 这种方法本质上是关键帧的局部删除, 由于是在已选择的关键帧中进行二次判断, 在减少关键帧数量的同时提高了建图和定位精度.

### 3 实验研究

所有实验均在 Ubuntu 12.04 环境下完成, 计算机 CPU 为 2.0 GHz Intel i5, 运行内存为 3 GB. 实验数据来自于 RGB-D 数据集——TUM RGB-D Benchmark<sup>[18]</sup> 和 NYU RGB-D Datasets<sup>[19]</sup>. 其中, coffee 来自于 NYU RGB-D Datasets, fr1\_xyz 和 long\_office\_household(loh) 来自于 TUM RGB-D Benchmark. 实验分别比较了常用关键帧选择算法(仅以  $D$ 、 $Inliers$  为关键帧选择依据)、改进关键帧选择算法(包含冗余关键帧)、改进关键帧选择算法(删除冗余关键帧)对 RGB-D SLAM 实时性、建图和定位精度以及关键帧数量的影响.

#### 3.1 RGB-D SLAM 实时性分析

实时性对任何 SLAM 来说都是重要的性能评价指标. 表 1 比较了 ORB、SIFT 和 SURF 特征在同一数据集下的实时性和匹配精度. 从表 1 可知, ORB 特征检测时间约为 SIFT 和 SURF 特征的 1/10, 且匹配时间远远少于 SIFT 和 SURF 特征. 同时, ORB 特征的匹配精度为 SIFT 和 SURF 特征的 2 倍以上. 显然 ORB 特征在检测和匹配速率上都比其他特征更快且精度更高.

表 1 ORB、SIFT 和 SURF 特征实时性和匹配精度比较

Tab. 1 Comparison of ORB, SIFT and SURF features in terms of real-time and matching precision

特征	匹配对	正确 匹配对	匹配 精度/%	检测 时间/ms	匹配 时间/ms
ORB	500	193	38.60	13.925	0.056
SIFT	961	141	14.67	345.989	44.641
SURF	1396	142	10.17	465.324	60.839

在使用 ORB 特征的基础上, 3 种关键帧选择算法对 RGB-D SLAM 实时性的影响结果如表 2 所示. 从表 2 可知, 无论是否删除冗余关键帧, 改进的关键帧选择算法都可以减少系统运作时间, 提高实时性. 特别是在场景 loh 中, 由于数据较多, 基于删除冗余关键帧的改进算法的实时性尤为显著.

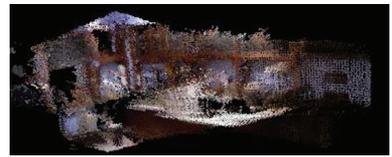
表 2 RGB-D SLAM 中 3 种关键帧选择算法实时性比较

Tab. 2 Comparison of three key-frame selection algorithms in terms of real-time in RGB-D SLAM

数据集	运作时间/ms		
	常用关键 帧选择 算法	改进关键帧 选择算法(包含 冗余关键帧)	改进关键帧 选择算法(删除 冗余关键帧)
fr1_xyz	84.33	82.64	75.85
coffee	93.15	92.51	81.63
loh	$635.08 \times 10^3$	$567.89 \times 10^3$	$417.74 \times 10^3$

#### 3.2 RGB-D SLAM 建图和定位精度分析

图 2 为 3 种关键帧选择算法构建的 3D 点云地图. 显然, 图 2(a)、(b)中存在大量漂移点, 场景中房屋、桌子和人的轮廓都不够清晰. 由于场景中动态行人, 前两种关键帧选择算法都不能很好地构建人所经过区域的局部地图. 与图 2(a)、(b)相比, 图 2(c)中的漂移点和冗余点较少, 房屋、桌子和人的轮廓很清晰. 并且人在移动过程中未被遮挡时的图像也可以清晰显示. 由此可见, 删除冗余关键帧改进算法构建的 3D 点云地图比其他两种算法精度更高.



(a) 关键帧选择算法



(b) 包含冗余关键帧改进算法



(c) 删除冗余关键帧改进算法

图 2 3 种关键帧选择算法构建的 3D 点云地图

Fig. 2 3D pointcloud map constructed by three key-frame selection algorithms

图 3(a)为 TUM 计算机视觉组利用运动视觉捕捉系统记录的摄像头在该场景下真实且连续

的轨迹,图 3(b)、(c)、(d)分别为 fr1\_xyz 场景下常用关键帧选择算法、删除冗余关键帧改进算法以及包含冗余关键帧改进算法计算的轨迹.各关键帧选择算法生成轨迹是非连续的,且与真实运

动轨迹(ground truth)间存在旋转和平移关系,所以并未将所有轨迹统一到一个坐标系中.可以看出,图 3(c)中删除冗余关键帧改进算法对应的运动轨迹奇点较少,与真实轨迹更加一致.

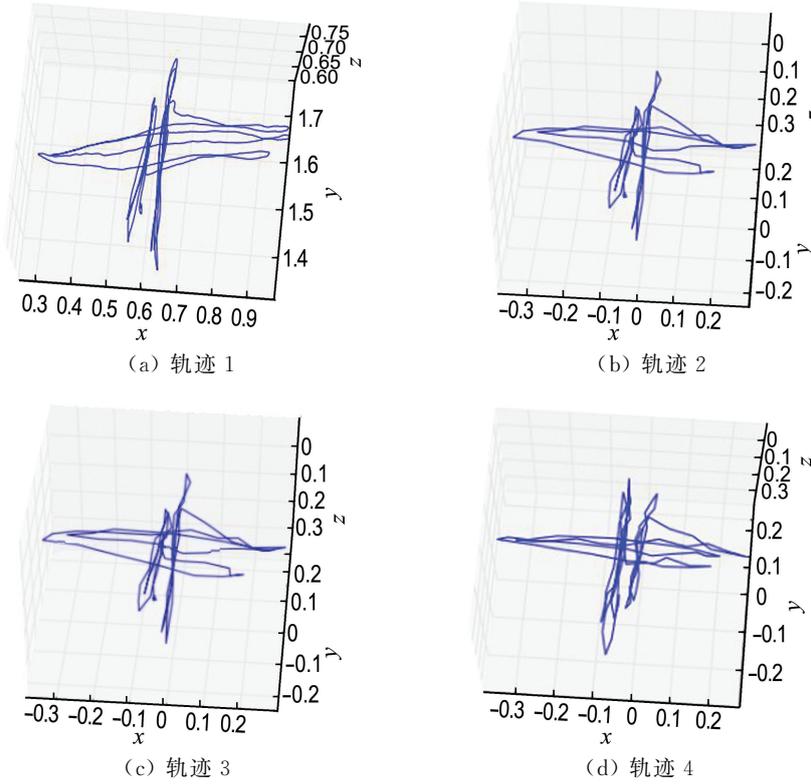


图 3 真实运动轨迹与 3 种关键帧选择算法的实际运动轨迹

Fig. 3 Ground truth and actual trajectory of three key-frame selection algorithms

对于定位精度<sup>[18]</sup>,采用实际运动轨迹与真实运动轨迹间的均方根误差(root mean square error,  $E_{rms}$ )进行比较.  $E_{rms}$  越小,精度越高,计算式如下:

$$E_{rms}(\hat{\mathbf{X}}, \mathbf{X}) = \sqrt{\frac{1}{n} \sum_{i=1}^n \|\text{trans}(\hat{\mathbf{x}}_i) - \text{trans}(\mathbf{x}_i)\|^2} \quad (2)$$

式中  $\hat{\mathbf{X}} = \{\hat{\mathbf{x}}_1, \dots, \hat{\mathbf{x}}_n\}$  表示根据关键帧选择算法生成的轨迹坐标,  $\mathbf{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_n\}$  为图 3(a)中真实轨迹坐标.

表 3 比较了 3 种关键帧选择算法对应的  $E_{rms}$ . 从表 3 可知,在不同实验数据下,改进关键帧选择算法(删除冗余关键帧)的  $E_{rms}$  最小,约为常用关键帧选择算法的 1/2,从而表明该算法的定位精度最高.

### 3.3 RGB-D SLAM 关键帧数量分析

本文通过对相邻两关键帧( $n=2$ )进行二次判

断来删除冗余关键帧.图 4 比较了不同数据集下 3 种关键帧选择算法计算的关键帧数量  $N$ . 很显然,改进关键帧选择算法(删除冗余关键帧)的关键帧数量远远小于其他两种算法.当 RGB-D SLAM 算法在大规模环境 loh 中使用时,这种差异尤为明显,其计算的关键帧数量约为常用关键帧选择算法的一半.

表 3 RGB-D SLAM 中 3 种关键帧选择算法定位精度的比较

Tab. 3 Comparison of localization accuracy of three key-frame selection algorithms in RGB-D SLAM

数据集	$E_{rms}/m$		
	常用关键帧选择算法	改进关键帧选择算法(包含冗余关键帧)	改进关键帧选择算法(删除冗余关键帧)
fr1_xyz	0.045	0.039	0.020
coffee	0.075	0.056	0.041
loh	0.996	0.873	0.592

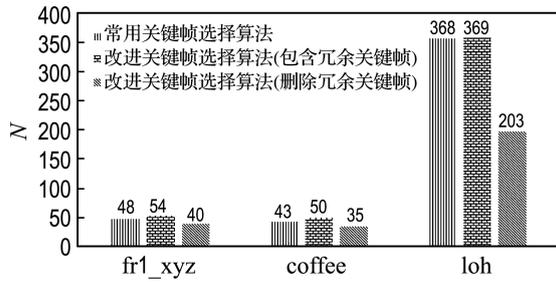


图 4 不同数据集下 3 种关键帧选择算法计算的关键帧数量

Fig. 4 The number of key-frame calculated by three key-frame selection algorithms under different datasets

### 3.4 实验结果分析

从上述实验中可以发现,引入删除冗余关键帧方法的改进关键帧选择算法在实时性、建图和定位精度方面性能较其他两种算法更好.在加入“当前环形闭合检测空闲或距  $Keyframe_L$  插入后又经过了  $z$  个帧”条件且未删除冗余关键帧时,改进算法中关键帧数量略多于常用关键帧选择算法中的数量,但是在加入删除冗余关键帧的方法后,改进算法中的关键帧的数量就会有所下降.特别是在大规模环境中,基于删除冗余关键帧的改进算法中的关键帧数量更是大幅下降.

实验过程中,当可调参数设置不准确时会出现地图重影、定位有奇异值等现象.对于不同的数据集,文中提到的这些参数最优取值范围往往需要通过大量的实验反复测试才能确定.

一般情况下,  $Inliers_{min}$ 、 $D_{min}$  和  $D_{max}$  可以通过相邻两帧图像之间的匹配情况以及算法实际所需精度来确定;  $Inliers_{min}$  越大,  $D_{min}$  和  $D_{max}$  的范围越小,关键帧的选取条件就越严格,在一定程度上会提高定位和建图的精度但也会增加系统的计算代价.

$\alpha$  表示  $Frame_{curr}$  与  $Frame_{curr-1}$  之间跟踪到的特征点个数,  $\alpha$  越大表明可跟踪的特征点个数越多,两帧图像匹配度越高.由于在  $Track(Frame_{curr-1}, Frame_{curr})$  之前已经使用常用关键帧选择算法进行了一次关键帧的筛选且还需为后续环节留有足够的候选关键帧,所以此时  $\alpha$  在 50~100 为宜.  $\beta$  表示  $Frame_{curr}$  与  $Keyframe_L$  之间匹配的特征点个数所占  $Keyframe_L$  特征点个数的比例.  $\beta$  越大表明  $Frame_{curr}$  与  $Keyframe_L$  之间的相关性越高.为了防止关键帧丢失以及满足最小视觉变化原则,  $\beta$  在 85%~95% 为宜.对于  $\alpha$ 、 $\beta$ , 均不能因精度要求而选择过大,这样不仅会导致候选关键帧数

量不足,还会增加计算量,降低算法实时性.设参数  $d$  表示帧间间隔空隙.  $d$  过小会频繁插入关键帧,在增加计算负荷的同时降低建图的精度甚至出现定位和环形闭合检测错误的现象;  $d$  过大则不能充分利用系统资源合理进行计算.一般情况下,  $d$  在 20~30 为宜.设参数  $k$  表示关键帧删除时需检测的关键帧个数,该参数可随着环境规模的增加适当增加.考虑到整个算法的实时性,  $k$  在 2~4 即可满足删除冗余关键帧的需求.

## 4 结 语

本文提出的改进关键帧选择算法不仅在一定程度上解决了常用关键帧选择算法鲁棒性较差、精度较低的问题,还通过删除冗余关键帧的方法减少了关键帧数量.整体上,本文提出的关键帧选择算法在及时、准确地插入关键帧的同时,提高了 RGB-D SLAM 算法的实时性、建图和定位精度.

## 参考文献:

- [1] GARCIA-FIDALGO E, ORTIZ A. Vision-based topological mapping and localization methods: A survey [J]. *Robotics and Autonomous Systems*, 2015, **64**:1-20.
- [2] CARLONE L, TRON R, DANILIDIS K, *et al.* Initialization techniques for 3D SLAM: A survey on rotation estimation and its use in pose graph optimization [J]. *Proceedings - IEEE International Conference on Robotics and Automation*, 2015, **2015-June**:4597-4604.
- [3] FUENTES-PACHECO J, RUIZ-ASCENCIO J, RENDÓN-MANCHA J M. Visual simultaneous localization and mapping: a survey [J]. *Artificial Intelligence Review*, 2012, **43**(1):55-81.
- [4] ENGEL J, SCHÖPS T, CREMERS D. LSD-SLAM: Large-scale direct monocular SLAM [J]. *Lecture Notes in Computer Science*, 2014, **8690 LNCS(part 2)**:834-849.
- [5] MUR-ARTAL R, MONTIEL J M M, TARDOS J D. ORB-SLAM: a versatile and accurate monocular SLAM system [J]. *IEEE Transactions on Robotics*, 2015, **31**(5):1147-1163.
- [6] ENDRES F, HESS J, STURM J, *et al.* 3D mapping with an RGB-D camera [J]. *IEEE Transactions on Robotics*, 2014, **30**(1):177-187.
- [7] KLEIN G, MURRAY D. Parallel tracking and mapping for small AR workspaces [C] // **2007 6th**

- IEEE and ACM International Symposium on Mixed and Augmented Reality, ISMAR. Piscataway: IEEE Computer Society, 2007:4538852.
- [8] LOWE D G. Distinctive image features from scale-invariant keypoints [J]. *International Journal of Computer Vision*, 2004, **60**(2):91-110.
- [9] BAY H, TUYTELAARS T, VAN GOOL L. SURF: Speeded up robust features [J]. *Lecture Notes in Computer Science*, 2006, **3951 LNCS**(1 of 4):404-417.
- [10] ROSTEN E, DRUMMOND T. Machine learning for high-speed corner detection [J]. *Lecture Notes in Computer Science*, 2006, **3951 LNCS**(1 of 4):430-443.
- [11] RUBLEE E, RABAU D V, KONOLIGE K, *et al.* ORB: An efficient alternative to SIFT or SURF [C] // **2011 International Conference on Computer Vision, ICCV 2011**. Piscataway: IEEE, 2011:2564-2571.
- [12] FISCHLER M A, BOLLES R C. 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.
- [13] DELLAERT F, KAESS M. Square root SAM: Simultaneous localization and mapping via square root information smoothing [J]. *International Journal of Robotics Research*, 2006, **25**(12):1181-1203.
- [14] FRESE U, LARSSON P, DUCKETT T. A multilevel relaxation algorithm for simultaneous localization and mapping [J]. *IEEE Transactions on Robotics*, 2005, **21**(2):196-207.
- [15] GRISSETTI G, STACHNISS C, GRZONKA S, *et al.* A tree parameterization for efficiently computing maximum likelihood maps using gradient descent [C] // **Robotics: Science and Systems**. Cambridge: MIT Press Journals, 2008:65-72.
- [16] KÜMMERLE R, GRISSETTI G, STRASDAT H, *et al.* G<sup>2</sup>o: A general framework for graph optimization [C] // **2011 IEEE International Conference on Robotics and Automation, ICRA 2011**. Piscataway: IEEE, 2011:3607-3613.
- [17] 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.
- [18] STURM J, ENGELHARD N, ENDRES F, *et al.* A benchmark for the evaluation of RGB-D SLAM systems [C] // **2012 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2012**. Piscataway: IEEE, 2012:573-580.
- [19] SILBERMAN N, FERGUS R. Indoor scene segmentation using a structured light sensor [C] // **2011 IEEE International Conference on Computer Vision Workshops, ICCV Workshops 2011**. Piscataway: IEEE, 2011:601-608.

## RGB-D SLAM algorithm based on improved key-frame selection

LI Yixing, LIU Shirong\*, ZHONG Chaoliang, WANG Jian

( School of Automation, Hangzhou Dianzi University, Hangzhou 310018, China )

**Abstract:** The key-frame selection is an important factor to improve the accuracy and real-time of visual SLAM algorithm. The key-frame selection is often based on the relative motion distance of the images. This method is simple and effective, but with poor real-time and robustness, and is easy to generate a large number of redundant key-frames. Aiming at the above problems, an improved key-frame selection algorithm is proposed. The algorithm incorporates the distance of relative motion of the frames, the feature points tracking and the minimum visual change to select a key-frame and delete the redundant key-frames. Based on this algorithm, combined the ORB (oriented FAST and rotated BRIEF) features with better direction and illumination invariant, the RGB-D SLAM algorithm is implemented. Experiments on RGB-D datasets indicate that the improved key-frame selection algorithm can be more accurate, timely in selecting key-frames, and improve the real-time and the accuracy of mapping and localization while reduce the redundant key-frames.

**Key words:** key-frames selection; redundant key-frames deletion; ORB feature; RGB-D SLAM