

文章编号: 2095-2163(2023)01-0084-08

中图分类号: TP242

文献标志码: A

基于 ORBSLAM2 改进的单线程双目 SLAM 系统

王劭靖

(北方工业大学 信息学院, 北京 100144)

摘要: ORBSLAM2 是一种稳定、鲁棒的实时 SLAM 系统,但其多线程结构给 CPU 带来了较重的负担,很难移植到嵌入式设备中。本文针对此问题对 ORBSLAM2 进行 3 处改进,将平滑的运动约束思想应用到运动模型的加速度中,使用加速运动模型预测相机的位姿,并添加环形匹配策略来剔除特征点的误匹配,快速、有效地确定良好的相机位姿初值和数据关联关系;将车联网领域中轨迹压缩的思想应用到 SLAM 中,提出一种基于开放窗口的关键帧聚类算法对位姿图中的位姿节点进行聚类,减少位姿图优化的规模;使用基于拉普拉斯分布的解耦优化三维空间点、相机的朝向与平移的方法代替局部捆绑调整(BA),提高优化的精度和速度。在 KITTI 和 New Tsukuba 数据集上的实验结果表明,改进后的 ORBSLAM2 能够以单线程的形式实时运行并且准确地估计相机的位姿。

关键词: ORBSLAM2; 开放窗口; 单线程; 拉普拉斯分布

Single-thread stereo SLAM system based on ORBSLAM2

WANG Shaojing

(School of Information, North China University of Technology, Beijing 100144, China)

[Abstract] ORBSLAM2 is a stable and robust real-time SLAM system, but its four-thread structure puts a heavy burden on the CPU and is difficult to be ported to embedded devices. This paper addresses this problem by making three improvements to ORBSLAM2, applying the idea of smooth motion constraints to the acceleration of the motion model, using the accelerated motion model to predict the camera's poses, and adding a ring matching strategy to eliminate the mis-matching of feature points to quickly and efficiently determine good initial values of camera poses and data association relationships; applying the idea of trajectory compression in the field of vehicle networking to SLAM. A key frame clustering algorithm based on open window is proposed to cluster the positional nodes in the positional map to reduce the scale of positional map optimization; a method based on the decoupling of Laplace distribution to optimize the orientation and translation of 3D spatial points and cameras is used instead of local bundle adjustment (BA) to improve the accuracy and speed of optimization. Experimental results on the KITTI and New Tsukuba datasets show that the improved ORBSLAM2 can run in real time as a single thread and accurately estimate the camera's poses.

[Key words] ORBSLAM2; open window; single thread; Laplace distribution

0 引言

视觉 SLAM (Simultaneous Localization And Mapping) 是计算机视觉和机器人技术界的一个热门研究课题。目前视觉 SLAM 技术广泛应用于航天机器人、AR/VR, 和自动驾驶等领域。

视觉 SLAM 系统主要分成前端和后端两大部分。前端又被称为视觉里程计, 主要分为基于直接法和基于特征点法两种; 后端主要分为基于滤波优化和基于非线性优化两种^[1]。近些年来, 基于直接法的代表性研究成果有 SVO 和 DSO, 但容易受光照

的影响导致位姿估计失败, 并且没有闭环检测功能, 使得估计的相机轨迹存在较大漂移, 在跟踪丢失后不容易重定位; 而基于特征点法的视觉 SLAM 系统对光照和动态物体不敏感并且稳定, 还可以利用基于描述子的闭环算法消除相机轨迹的漂移, 减少累计误差, 因此基于特征点法的视觉 SLAM 系统是目前主流且成熟的视觉方案, 并且大多使用基于非线性优化的后端。PTAM 是首个基于特征点法的实时视觉 SLAM 系统, 并且首创了前端和后端并行的多线程结构^[2]。随后出现的许多优秀的视觉 SLAM 系统都是基于 PTAM 进行改进的, 其中 ORBSLAM2 被

作者简介: 王劭靖(1995-), 男, 硕士研究生, 主要研究方向: 视觉 SLAM。

收稿日期: 2022-04-06

业界人士认为是当前最稳定的视觉 SLAM 系统,可应用于多种场景,在视觉 SLAM 领域具有里程碑的意义。

ORB_SLAM2 使用了四线程结构。跟踪线程主要对提取的特征点利用描述子进行匹配,从而构造数据关联,并根据数据关联关系对相机位姿实时地进行估计。在当前帧确定为关键帧后,建图线程使用局部 BA 优化共视图 (Co-visibility graph) 中的关键帧和三维空间点。闭环线程中优秀的闭环检测算法和基于本质图 (Essential graph) 的位姿图优化保证了 ORB_SLAM2 能够有效地减少累计误差。此外,全局 BA 单独占用一个线程同时优化所有的关键帧和三维空间点。因此,ORB_SLAM2 比其他现有视觉 SLAM 系统具有更好的准确性和鲁棒性,但其四线程结构给 CPU 带来了较重的负担,很难将其移植到嵌入式设备上。此外,航天设备上需要多种技术共同协作,SLAM 的资源占用率的减少不仅能够降低硬件成本,还能满足其他技术对资源的需求。因此,本文将对 ORB_SLAM2 进行改进,使其能够以单线程的形式在 CPU 上实时运行并且准确地估计相机位姿。

1 相关工作

视觉 SLAM 系统前端主要负责数据关联与位姿估计。良好的位姿初值是建立准确数据关联的前提,有效的运动模型能够提供一个良好的位姿初值,从而加速特征匹配的速度和精度,并且加快位姿估计的速度。常见的运动模型通常在相机相邻两帧之间应用平滑的运动约束^[3],即假设前后两帧的运动速度相似,DP-PTAM、DT-SLAM 和 ORB_SLAM2 都利用恒速运动模型假设相机前后两帧的运动速度相同,进而预测相机的位姿初值。SVO 和 LSD-SLAM 直接令相机当前时刻的位姿等于上一时刻的位姿,即假设当前帧的运动速度为 0。但在实际应用中,相机的运动速度是经常变化的,不符合上述的假设,因此本文将相机的运动假设为速度不断变化的加速运动,将平滑运动约束的思想应用到前后帧的加速度上,假设前后帧的加速度相同来计算当前时刻的速度,最终预测出当前时刻的位姿初值,以提高数据关联的效率和正确性。

由于 BA 同时优化大量的三维空间点坐标和相机位姿,因此计算复杂度高导致运算时间过长,很难实现快速而准确的收敛^[4],通常需要开启多个线程来分担计算量以满足实时性的要求。Chatterjee^[5]等

人提出了一种基于 L_1 范数的旋转平均 (Rotation averaging) 的方法能够鲁棒、快速地单独优化相机的朝向。近些年来许多研究都证实了基于旋转平均的解耦合优化方法比 BA 的效率和精度更高,能够快速、精确地优化出相机的位姿。Fan^[6]等人提出了将三维空间点的位置、相机的朝向和平移进行解耦合优化的方法来提高优化的效率和精确性,并且假设噪声服从拉普拉斯分布,以提高对存在大噪声数据优化的鲁棒性。本文采用基于平均的解耦合优化方法代替 ORB_SLAM2 中的 Local BA,以提高优化的效率、鲁棒性和精确性,并减少 SLAM 的线程数量。

位姿图优化是 SLAM 进行全局优化的重要手段,能够有效减少相机轨迹的漂移,从而减少累积误差。但随着相机的不断探索,位姿图的规模会越来越大,使得优化效率受限。Kretzschmar^[7]等人提出了基于信息论的方法,减少位姿图优化的规模,通过计算互信息和信息增益的期望挑选出需要边缘化的节点,并提出了一种基于 Chow-Liu 树的近似边缘化方法保证位姿图的稀疏性,但理论复杂且实现较困难;Johannsson^[8]、Zeng^[9]等人认为当连续帧连续检测到闭环时,说明相机正重新经过之前探索过的区域,就将连续检测到闭环的多个帧视为冗余,并将多个冗余帧和回环帧之间的约束进行合并,直接将冗余节点去除,不进行边缘化操作,但不适用于大场景;Latif^[10]等人认为在二维平面上运动且不能原地旋转的机器人在做直线运动时朝向是几乎不变的,因此直线运动中的约束对位姿图优化提供的是线性约束,于是就将直线运动中的多个约束合并为一个约束,只对位姿图中朝向改变较大的位姿节点进行优化,但对相机的运动条件限制严格,不适用于三维场景中能够自由转动的相机;车联网领域,Keogh^[11]等人提出的开放窗口算法是一种快速压缩轨迹数据的简单方法,该算法将车辆轨迹根据弯曲程度用多个线段子轨迹表示。本文将车联网领域中轨迹压缩的思想应用到 SLAM 中,并对 Latif 等人的方法进行改进,以此只优化朝向变化较大的位姿节点,减少位姿图优化的规模,并且适用于在三维场景中自由转动的相机。

2 具体改进

本文提出的 SLAM 系统是基于 ORB_SLAM2 的框架改进的,本文 SLAM 系统的流程图如图 1 所示,主要改进:

(1) 使用加速运动模型预测相机的位姿,并结

合环形匹配策略提高特征匹配的精度和速度;

(2)在当前帧确定为关键帧后,利用基于开放窗口的关键帧聚类算法对其进行聚类,减少位姿图优化的规模,加速位姿图优化的速度;

(3)在噪声服从拉普拉斯分布假设的前提下,使用解耦合优化三维点空间、相机的朝向和平移的方法代替 Local BA,以此提高优化的速度和精度。

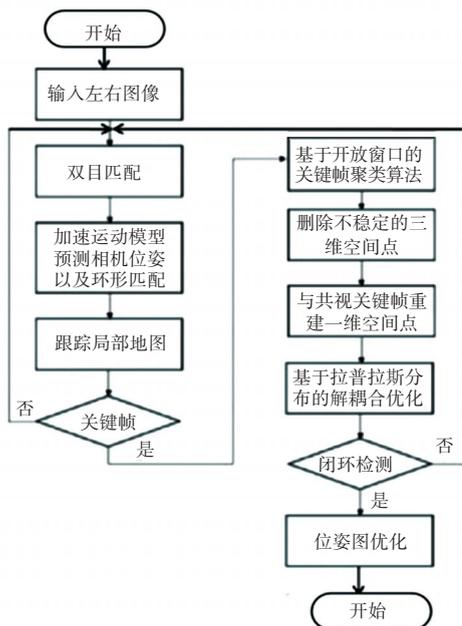


图1 本文 SLAM 系统的流程图

Fig. 1 The flow chart diagram of our SLAM system

2.1 基于平滑运动约束的加速运动模型

良好的位姿初值不仅仅能提高数据关联的正确性,还可以减少优化的迭代次数,加速收敛速度。本文考虑到相机的运动大部分情况是速度不断变化的加速运动,匀速运动的情况较少,并且匀速运动和减速运动也分别是一种加速度为 0 和加速度为负数的特殊加速运动。因此本文利用加速运动模型估计相机位姿的初值,并将平滑运动约束的思想应用到运动模型的加速度上,假设相邻两帧的加速度相同,则有式(1):

$$V_t * V_{t-1}^{-1} = V_{t-1} * V_{t-2}^{-1} \quad (1)$$

其中, V_t 、 V_{t-1} 和 V_{t-2} 分别代表 t 帧、 $t-1$ 帧和 $t-2$ 帧的速度, V_{t-1}^{-1} 为 V_{t-1} 的逆。

$t-1$ 帧和 $t-2$ 帧的速度,式(2)和式(3):

$$V_{t-1} = T_{t-1} * T_{t-2}^{-1} \quad (2)$$

$$V_{t-2} = T_{t-2} * T_{t-3}^{-1} \quad (3)$$

其中, T_{t-1} 、 T_{t-2} 、 T_{t-3} 分别代表 $t-1$ 帧和 $t-2$ 帧和 $t-3$ 帧的位姿。

因此 t 帧的速度和预测的位姿可由以上公式推

导而出,式(4)和式(5):

$$V_t = V_{t-1} * V_{t-2}^{-1} * V_{t-1} \quad (4)$$

$$T_t = V_t * T_{t-1} \quad (5)$$

由此能够根据 t 帧的预测位姿将 $t-1$ 帧观测到的三维空间点投影至 t 帧进行匹配。为了使特征匹配更可靠,本文加入了环形匹配策略。在 ORBSLAM2 中,每一帧左右图像的特征匹配关系是在每一帧初始化时确定的,因此在通过投影得到前后帧左图像中特征点的对应关系后,可直接得到前后帧右图像中特征点的对应关系,利用描述子验证环形是否闭合,最终确定匹配成功的特征点,如图 2 所示。特征匹配结果对比如图 3 所示,可以看出本文的特征匹配数量相比于 ORBSLAM2 少,但准确度更高,进一步剔除了更多的误匹配。

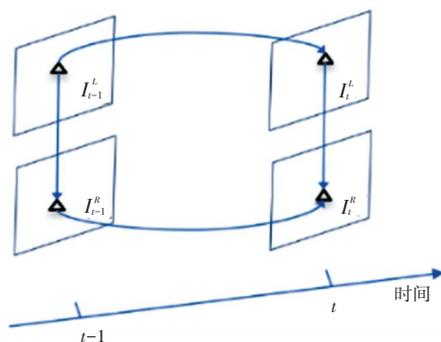
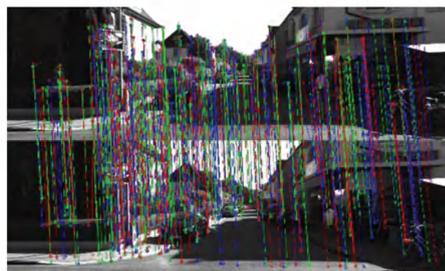
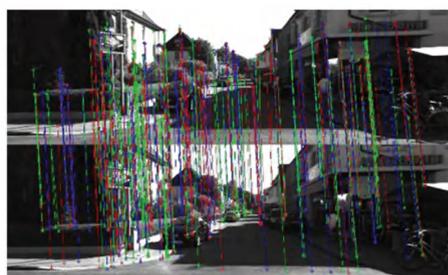


图2 环形匹配示意图

Fig. 2 Schematic diagram of circle matching



(a) ORBSLAM2 特征匹配结果示意图



(b) 本文特征匹配结果示意图

图3 特征匹配结果对比示意图

Fig. 3 Schematic diagram of feature matching results comparison

2.2 基于开放窗口的关键帧聚类算法

为了挑选出运动过程中朝向变化较大的位姿节点,本文提出了基于开放窗口的关键帧聚类算法。该算法首先对位姿图中的位姿节点在 \mathbb{R}^3 空间中聚类,以此将位姿图划分为多个线段子轨迹,每个线段子轨迹都是一段直线运动,每个直线运动的尾结点和下一个直线运动的首节点之间会发生较大的朝向变化。此外,由于相机在直线运动过程中也会出现朝向变化较大的情况,因此每一段在 \mathbb{R}^3 空间中确定的子轨迹需要继续在 $SO(3)$ 空间中进行聚类,以此有效挑选出朝向变化较大的节点进行优化,减小位姿图优化规模。

\mathbb{R}^3 空间中关键帧聚类示意如图 4 所示,在当前帧确定为关键帧时,令其为 $x_{current}$,窗口中的位姿节点为 x_{start} 和 x_{end} 之间的所有位姿节点,根据 x_{start} 和 x_{end} 可确定直线 L ,由此可以计算窗口内所有位姿节点相对直线 L 的偏移量,即位姿节点到直线 L 的垂直距离,即式(6):

$$distance(x_i, L) = \frac{\| (t_i - t_{start}) \times (t_{end} - t_{start}) \|}{\| t_{end} - t_{start} \|} \quad (6)$$

其中, $t \in \mathbb{R}^3$, 代表位姿节点的平移部分。

当偏移距离最大的 x_i 的偏移距离大于阈值时,将 x_{start} 和 x_i 之间的所有节点聚为一类,表示在 \mathbb{R}^3 空间中的聚类结果,用 x_{start} 和 x_i 代表这段子轨迹,并更新窗口,即令 x_{start} 为 x_{i+1} , 令 x_{end} 为 $x_{current}$ 。

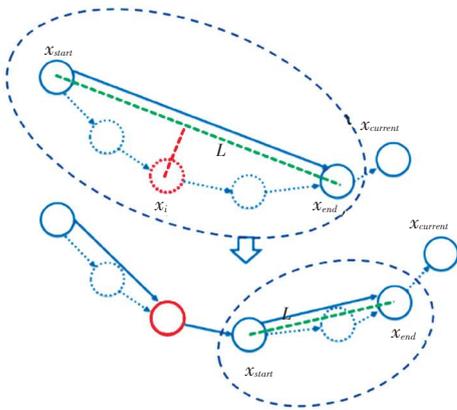


图 4 \mathbb{R}^3 空间中关键帧聚类示意图

Fig. 4 Schematic diagram of keyframe clustering in \mathbb{R}^3

每一段在 \mathbb{R}^3 空间中确定的子轨迹再分为两步在 $SO(3)$ 空间中进行聚类:

第一步,在 x_{start} 与 x_i 构成的子轨迹中判断相邻位姿节点之间的朝向变化,当 x_j 和 x_{j+1} 朝向变化大于阈值时,就将这段子轨迹继续划分,即 x_{start} 和 x_j 之间的所有位姿节点聚为一段子轨迹,并继续判断后

面相邻位姿节点之间的朝向变化,如图 5 所示;

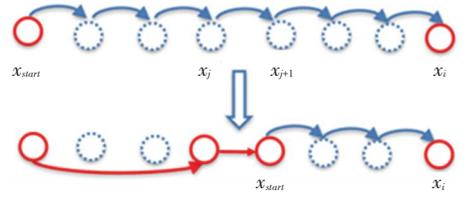


图 5 $SO(3)$ 空间中基于相邻关键帧朝向变化的关键帧聚类示意图
Fig. 5 Schematic diagram of keyframe clustering based on the orientation change of adjacent keyframes in $SO(3)$

第二步,考虑到存在相邻帧之间朝向变化不大而不相邻两帧之间朝向变化较大的情况,本文将第一步中的子轨迹在 $SO(3)$ 空间中进行第二次聚类,即当 x_{start} 与 x_k 之间朝向变化大于阈值时,就将 x_{start} 和 x_j 这段子轨迹继续划分,即 x_{start} 和 x_{k-1} 之间的所有位姿节点聚为一段子轨迹,并继续判断后面不相邻位姿节点之间的朝向变化,如图 6 所示。具体的基于开放窗口的关键帧聚类算法如算法 1 所示,其中 $Dthreshold$ 代表在 \mathbb{R}^3 空间中进行聚类的偏移距离阈值, $thetaThreshold$ 代表在 $SO(3)$ 空间中进行聚类的角度阈值。

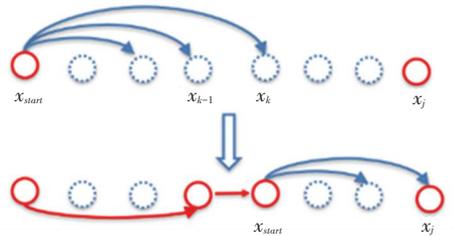


图 6 $SO(3)$ 空间中基于不相邻关键帧朝向变化的关键帧聚类示意图
Fig. 6 Schematic diagram of keyframe clustering based on the orientation change of non-adjacent keyframes in $SO(3)$

在 KITTI 07 序列上应用该聚类算法的结果示意图如图 7 所示。图 7 中的大节点是朝向变化较大的节点,即需要优化的位姿节点,可以看出本文提出的基于开放窗口的关键帧聚类算法能够有效减少位姿图优化的规模。针对其余未进行位姿图优化的节点位姿,本文使用位姿传播的方式更新。

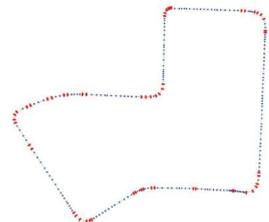


图 7 基于开放窗口的关键帧聚类算法在 KITTI07 上的聚类结果示意图
Fig. 7 Schematic diagram of the clustering results of the keyframe clustering algorithm based on the open window on KITTI07

2.3 解耦合优化方法

为了在 SLAM 后端避免复杂的 BA 优化,本文在假设噪声服从拉普拉斯分布的前提下构建 L_1 范数误差,使用解耦合优化三维空间点、相机朝向和平移的方法代替 Local BA。

得益于根据数据关联建立的共视图,一个三维空间点在多个关键帧中的观测能够快速被找到。因此,三维空间点在多个关键帧的观测根据双目视差能够计算出多个坐标,如图 8 所示。本文采用 \mathbb{R}^3 上的 IRLS (Iterative Reweighted Least Squares) 算法优化三维空间点坐标为

$$P^* = \operatorname{argmin}_{\hat{P}} \sum_{i=1}^n \|P_i - \hat{P}\|_1 \quad (7)$$

其中, \hat{P} 代表三维空间点的坐标初值,本文取多个坐标中的任意一个坐标作为初值, P^* 代表三维空间点优化后的坐标。

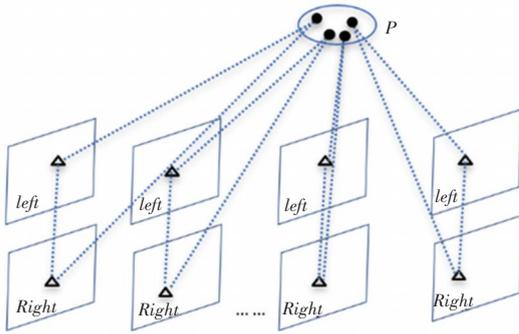


图 8 利用双目视差多次重建三维空间点示意图

Fig. 8 Schematic diagrams of 3D space points are reconstructed multiple times according to binocular disparity

对于相机朝向的优化,本文利用数据关联关系快速得到当前关键帧与多个共视关键帧的特征匹配关系,从而使用五点法求解出当前关键帧和多个共视关键帧之间本质矩阵(Essential matrix),因此可得到当前关键帧的多个朝向,使用 $SO(3)$ 上的 IRLS (Iteration Reweight Least Square) 算法优化相机的朝向为

$$R^* = \operatorname{argmin}_{\hat{R}} \sum_{i=1}^m \|R_i - \hat{R}\|_1 \quad (8)$$

其中, \hat{R} 代表相机朝向的初值,本文取多个朝向中的任意一个朝向作为初值, R^* 代表优化后的相机朝向。

最后利用优化后的三维空间点和相机的朝向根据重投影误差优化相机的平移为

$$t^* = \operatorname{argmin}_{\hat{t}} \sum_{i=1}^k \|u_i - \pi(RP_i + \hat{t})\|_1 \quad (9)$$

其中, u_i 代表三维空间点 P_i 在当前关键帧上的二维观测点坐标, $\pi(RP_i + t)$ 代表三维空间点 P_i 对应在当前帧上的投影。

3 实验结果与分析

本文的实验环境为 Ubuntu 18.04、配置为 Inter Core i7-5500 CPU、主频 2.4 GHz、8 G RAM。为了检测本文 SLAM 算法的性能,本文从精度和速度两个方面在 KITTI 数据集以及 New Tsukuba 数据集上进行测试评估。KITTI 数据集采集了大型室外场景下的视频序列,并提供了真实轨迹的数据; New Tsukuba 数据集是由计算机合成的静态室内小场景。本文选择与目前开源的 Viso2 和 ORBSLAM2 算法进行对比,使用均方根误差(RMSE)来衡量相机真实位姿与估计的位姿之间的误差。本文在算法 1 中使用的偏移距离阈值为 0.5,角度阈值为 0.2 rad。

Viso2、ORBSLAM2 和本文方法在 KITTI 数据集上和 New Tsukuba 数据集的测试结果见表 1,可以看出本文方法的精度比 ORBSLAM2 的精度与 Viso2 的精度高,并且由于 Viso2 没有闭环检测功能,在具有闭环的序列中表现差距更加明显。

表 1 Viso2、ORBSLAM2 和本文方法在 KITTI 和 New Tsukuba 数据集上的 RMSE (m) 对比

Tab. 1 RMSE (m) comparison of Viso2, ORBSLAM2 and our method on KITTI and New Tsukuba

序列	Viso2	ORBSLAM2	Our Method
00	64.285	6.78	5.358
01	130.598	23.322	22.01
02	76.311	13.008	11.466
03	14.224	1.975	1.79
04	3.653	0.862	0.689
05	25.799	1.537	1.461
06	7.968	1.906	1.774
07	12.057	0.989	0.879
08	65.274	13.068	12.088
09	53.517	7.034	4.981
New Tsukuba	0.331	0.107	0.092

本文算法在精度方面的优势主要得益于加速运动模型以及环形匹配策略提高了数据关联的准确性,尽管在数据关联过程中错误的关联并不能完全消除,但基于构建 L_1 范数误差的解耦合优化方法提高了对存在异常值数据进行的优化鲁棒性。

ORBSLAM2 与本文方法在 New Tsukuba 数据集上处理每一帧的时间对比图如图 9 所示,其中细折线代表 ORBSLAM2 处理每一帧的时间,粗折线代表本文方法处理每一帧的时间,可以看出本文方法处理每一帧的时间整体上要低于 ORBSLAM2 处理每

一帧的时间。本文的 SLAM 算法与 ORBSLAM2 在 New Tsukuba 数据集上的计算效率和线程数量对比见表 2, 可以看出本文方法的计算效率高于 ORBSLAM2, 并且能够以单线程的形式运行。

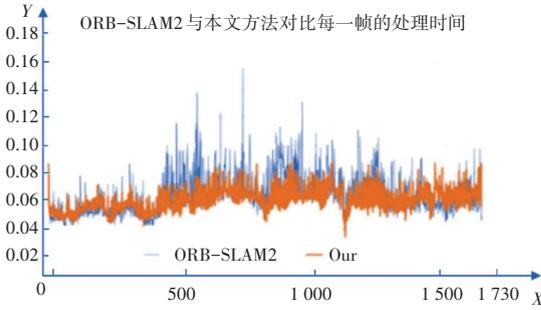


图 9 ORBSLAM2 与本文方法在 New Tsukuba 数据集上处理每一帧的时间对比图

Fig. 9 Time comparison of ORBSLAM2 and our method on the New Tsukuba dataset

表 2 本文算法与 ORBSLAM2 在 New Tsukuba 数据集上的计算效率和线程数量对比

Tab. 2 Comparison of computational efficiency and number of threads between our algorithm and ORBSLAM2 on the New Tsukuba dataset

算法	Mean[ms/frame]	Median[ms/frame]	Threads
ORBSLAM2	64.35	61.51	4
Our	60.37	59.57	1

本文算法对比于 ORBSLAM2 在计算效率和线程数量方面的优势主要归因于如下几点:

(1) 加速运动模型预测的良好位姿初值和较为准确的数据关联使得位姿估计过程中的仅运动 BA (motion-only BA) 计算的梯度更为准确, 能够使位姿估计过程更快地收敛;

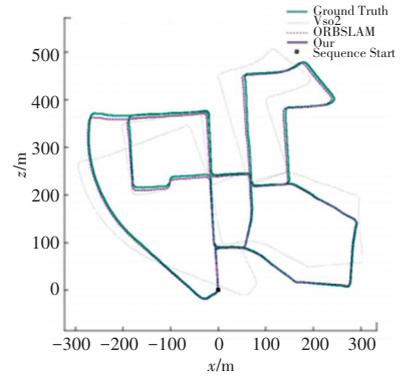
(2) 解耦合的优化方法计算复杂度比 BA 低, 由此减少了优化时间;

(3) 基于开放窗口的关键帧聚类算法能够明显减少位姿图优化的规模, 使得全局优化的效率得以提升;

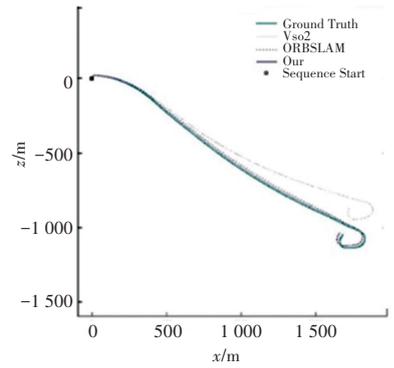
(4) 由于位姿估计过程并不需要太多数量的三维空间点, 过多的三维空间点对位姿估计的精度提升并不明显, 还会增加计算的复杂性。因此本文只利用当前帧的共视关键帧所观测到的三维空间点对当前帧进行位姿估计, 相比 ORBSLAM2 利用当前帧的共视关键帧以及共视关键帧的共视关键帧所观测到的三维空间点对当前帧进行位姿估计的方法, 能够有效减少不必要的计算量, 并且保证位姿估计的精度。

在 KITTI 数据集和 New Tsukuba 数据集上本文的方法、Viso2、ORBSLAM2 的轨迹和真实轨迹对比

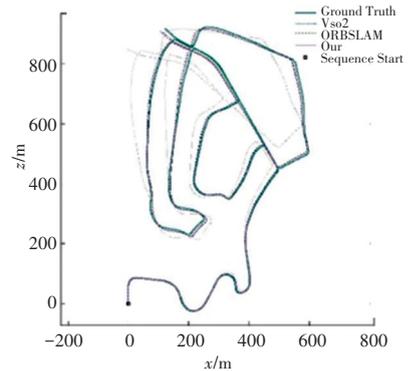
如图 10 所示, 其中粗实线为真实轨迹, 细实线为本文方法估计的轨迹, 粗虚线为 ORBSLAM2 估计的轨迹, 细虚线为 Viso2 估计的轨迹, 可以看出本文方法估计出的轨迹更接近于真实轨迹。



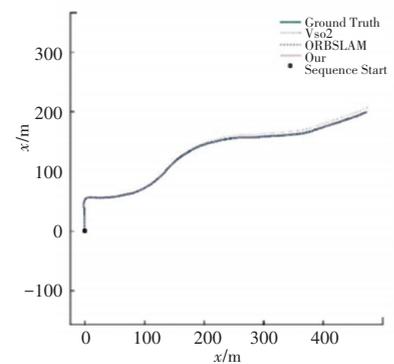
(a) KITTI00



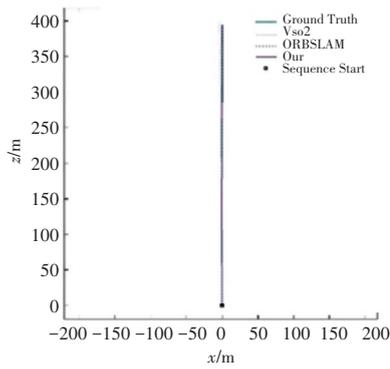
(b) KITTI01



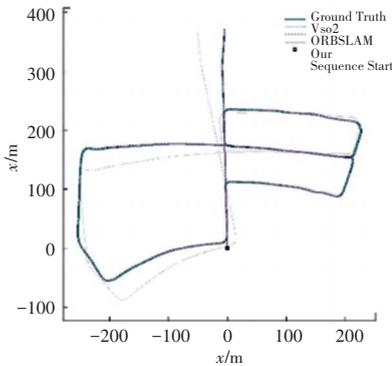
(c) KITTI02



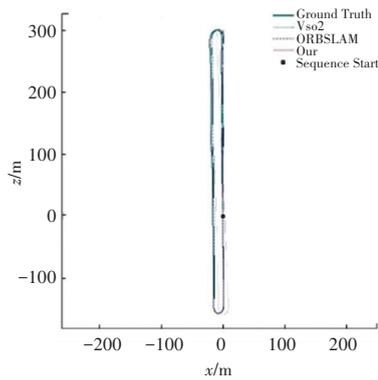
(d) KITTI03



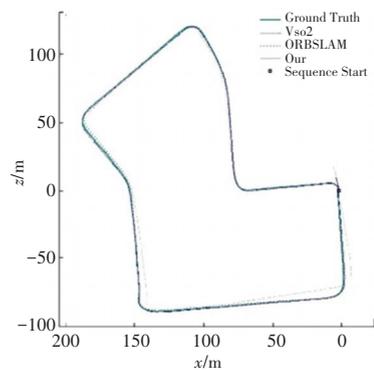
(e) KITTI04



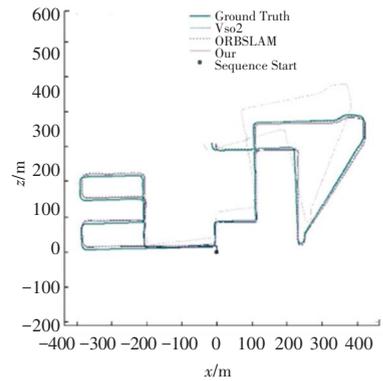
(f) KITTI05



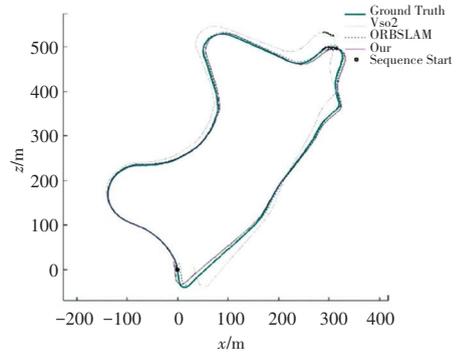
(g) KITTI06



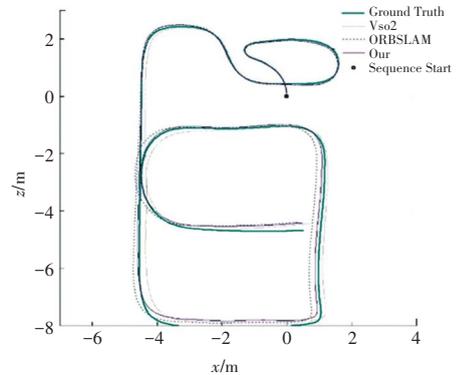
(h) KITTI07



(i) KITTI08



(j) KITTI09



(k) New Tsukuba

图 10 KITTI 数据集和 New Tsukuba 数据集上本文的方法、Viso2、ORBSLAM2 的轨迹和真实轨迹对比图

Fig. 10 Comparison of trajectories and real trajectories of our method、Viso2、ORBSLAM2 on the KITTI dataset and the New Tsukuba dataset

4 结束语

本文主要针对 ORBSLAM2 进行 3 方面改进:

(1) 使用加速运动模型预测相机的位姿,并结合环形匹配策略提高了数据关联的准确性和位姿优化的收敛速度;

(2) 基于开放窗口算法提出了一种简单有效的关键帧聚类算法,从而减少位姿图优化的规模;

(3) 使用基于拉普拉斯分布的解耦合优化方法

(下转第 99 页)