

基于改进VINS-Mono算法的状态估计方法

王海芳, 乔鼎杰, 吴天浩, 黄鹏

(东北大学秦皇岛分校 控制工程学院, 河北 秦皇岛 066004)

摘要: 针对传统的即时定位并建图(simultaneous localization and mapping, SLAM)算法在位姿识别过程中对精度的要求,在VINS-Mono(visual inertial system-Mono)的前端部分,增加了逆向光流法;针对SLAM算法对实时性的要求,在VINS-Mono的后端滑窗优化方法中,融合了一种边缘化优化算法,再对除相机位姿的部分进行边缘化,然后边缘化相机的位姿部分,从而加速边缘化的过程.再使用EuRoc(European robotics challenge)数据集进行实验,结果发现针对前端的改进策略,精度提升不明显,并分析了原因;针对后端的改进策略,改进算法的边缘化时间平均减少了25.9%,又对比了改进算法与源码的轨迹精度,发现误差可控.最后验证了对VINS-Mono后端的改进策略在实时性上具有优越性.

关键词: 状态估计方法;滑窗;边缘化;实时性;ROS(robot operating system)机器人仿真平台

中图分类号: TP 301 **文献标志码:** A **文章编号:** 1005-3026(2024)09-1287-07

State Estimation Method Based on Improved VINS-Mono Algorithm

WANG Hai-fang, QIAO Ding-jie, WU Tian-hao, HUANG Peng

(School of Control Engineering, Northeastern University at Qinhuangdao, Qinhuangdao 066004, China. Corresponding author: WANG Hai-fang, E-mail: hfwang0335@126.com)

Abstract: Aiming at the requirement of precision in pose recognition process of traditional SLAM (simultaneous localization and mapping), the reverse optical flow algorithm is added to the front-end part of VINS-Mono. To meet the real-time requirement of SLAM, a marginal optimization algorithm is integrated into the back-end sliding window optimization method of VINS-Mono, the parts excluding camera pose are marginalized firstly, and then pose parts of cameras are marginalized, to accelerate marginalization process. And experiments are carried out using EuRoc(European robotics challenge)datasets. The results show that accuracy is improved slightly by improvement strategy of the front end, and the possible reasons are analyzed. For the improvement strategy of the back end, it is found that compared with the source code, the time used for the marginalization of the improved algorithm is reduced by 25.9% on average, and then compare their trajectory accuracy. It is found that the error is controllable. Finally, it is verified that the improved strategy for the back-end of VINS-Mono is superior in real-time performance.

Key words: state estimation method; sliding window; marginalization; real-time performance; ROS robot simulation platform

在社会不断进步、科技持续发展的背景下,移动机器人在生活、生产中的应用也越来越广泛,其状态估计问题受到了广泛的关注.即时定位并建图(SLAM)算法成为了实现机器人独立定位和导航的关键. Davison等^[1]提出的Mono-

SLAM是首个即时单目视觉SLAM算法,该算法在前端追踪离散的特征点,后端使用拓展卡尔曼滤波,然后更新相机目前的状态和路标点的平均值、方差. Klein等^[2]提出了PTAM(parallel tracking and mapping),在后端使用非线性优化替代了拓

展卡尔曼滤波,使跟踪与建图得以同时进行. LSD-SLAM (large scale direct monocular-SLAM) 是 Engel 等^[3-4]提出的,在前端视觉里程计部分使用直接法,避开了计算特征点的过程,还可以绘制半稠密地图,但该算法同时继承了直接法的缺陷,对相机的内部参数以及曝光与否非常敏感,而且在相机快速运动时容易出现特征点丢失的情况. Forster 等^[5]提出了一种基于稀疏直接法的视觉里程计 SVO (semi-direct visual odometry),速度极快,即使在低端计算平台上也能达到实时性,并且使用了深度滤波器的概念. DTAM (dense tracking and mapping)^[6]、DSO (direct sparse odometry)^[7]、DVO (dense visual odometry)^[8]等都是基于直接法的单目或双目 SLAM 方案. ORBSLAM (oriented fast and rotated brief-SLAM)^[9]支持单目、双目、RGB-D (red, green, blue, depth) 三种模式,使用跟踪、局部 BA (bundle adjustment) 优化、回环检测 3 个线程,保证轨迹与地图的全局一致性.但是纯视觉在相机曝光过度、相机运动过快以及图像过于模糊的情况下效果不太理想,会出现求解位姿时精度较低等现象,用惯性测量单元 IMU (inertial measurement unit) 与之融合,可以有效解决上述问题,代表算法有: MSCKF (multi-state constraint Kalman filter)^[10]、ROVIO (robust visual inertial odometry)^[11]、MSF (multi-sensor fusion)^[12]、OKVIS (open keyframe-based visual-inertial SLAM)^[13]、VINS (visual inertial system)^[14]等. IMU 与相机可以互相补充不足,相机可以纠正 IMU 的累计漂移,在相机快速运动时,高频率的 IMU 能够减少相机受到的影响. VINS 使用一种紧耦合的 VIO (visual-inertial odometry)^[15],构建 IMU 与相机共同的运动、观测方程,再进行状态估计.为了进一步提高算法的精度以及实时性,本文在 VINS-Mono 的基础上,前端增加了逆向光流匹配,希望增加轨迹精度;后端对边缘化中的残差信息进行了优化,并对信息矩阵分两步进行边缘化,从而加快边缘化的速度.

1 算法介绍

1.1 VINS 的前端

VINS-Mono 分为测量预处理模块、初始化模块、VIO 模块以及全局位姿图优化模块.在测量预处理模块时,可以进行特征点的检测与光流

跟踪.先剔除特征点中位于图像边界之外的点,再通过对极约束的方式剔除质量不好的点.然后令特征点的分布均匀化,对每一个特征点进行去畸变操作.由于剔除了一些外点,所以需要补充一些新的特征点.之后,对于 IMU 信息,进行相应的预积分,将得到的结果进行视觉惯性对齐,这样就保证了每一帧图像都有若干 IMU 信息与其匹配.因为单目相机是没有尺度的,所以根据 IMU 信息与图像信息对齐后的测量值,通过非线性优化的方法解出特征点的深度,这样就完成了初始化模块.

1.2 光流法

光流法是一种描述像素随时间在图像之间运动的方法,稀疏光流计算图像部分像素的运动,稠密光流计算图像内全部像素的运动. VINS 前端采用的是稀疏光流法,以 Lucas-Kanade 光流^[16]为代表.设在 $t=1$ s 时的图像是运动前的图像,其中 1 个图像块 $T(\mathbf{x})$ 称为模板图像,其像素位置为 $\mathbf{x}=[x,y]^T$. 在相机运动之后,模板图像在 $t=2$ s 时的新位置,称为输入图像,其像素位置更新为 $W(\mathbf{x}; \mathbf{p})$,所以此时输入图像的像素为 $I(W(\mathbf{x}; \mathbf{p}))$. 例如,对于平面图像, $W(\mathbf{x}; \mathbf{p})$ 可被定义为 $W(\mathbf{x}; \mathbf{p}) = \begin{pmatrix} x+p_1 \\ y+p_2 \end{pmatrix}$, 其中参数向量 $\mathbf{p}=(p_1,p_2)^T$ 表示像素在 x,y 轴的运动,也就是光流,是待求解的部分.

首先需要引入光流的基本假设:灰度不变,即某个特征点的像素灰度值,在各图像中固定不变.在实际中可能会由于物体的材质不同,使得像素出现高光和阴影部分;有时相机会自动调整曝光参数,使得图像整体变亮或变暗,此时灰度不变假设都是不成立的.但是 VINS 获取图像的频率是 30 Hz,此时 2 幅图像之间的间隔是 0.03 s,可以认为 2 幅图像在同一点的像素灰度不变.假设在 1 个窗口 A 内的像素都具有相同的运动,即都满足灰度不变假设,那么 $I(W(\mathbf{x}; \mathbf{p}))$ 与 $T(\mathbf{x})$ 就应该非常接近,因此得到

$$\sum_{\mathbf{x} \in A} [I(W(\mathbf{x}; \mathbf{p})) - T(\mathbf{x})]^2. \quad (1)$$

假设当前的像素运动 \mathbf{p} 已知,然后迭代求解参数 \mathbf{p} 的增量 $\Delta \mathbf{p}$,因此需要最小化的目标函数为

$$\sum_{\mathbf{x} \in A} [I(W(\mathbf{x}; \mathbf{p} + \Delta \mathbf{p})) - T(\mathbf{x})]^2. \quad (2)$$

其中优化的变量是 $\Delta \mathbf{p}$. 然后迭代更新参数 \mathbf{p} :

$$\mathbf{p} \leftarrow \mathbf{p} + \Delta \mathbf{p}. \quad (3)$$

3 个公式不断迭代,直到参数 \mathbf{p} 收敛为止.

对 $I(\mathbf{W}(\mathbf{x}; \mathbf{p} + \Delta\mathbf{p}))$ 用泰勒公式进行一阶展开,从而将目标函数优化为

$$\sum_{\mathbf{x} \in \mathcal{A}} \left[I(\mathbf{W}(\mathbf{x}; \mathbf{p})) + \nabla I \cdot \frac{\partial \mathbf{W}}{\partial \mathbf{p}} \cdot \Delta\mathbf{p} - T(\mathbf{x}) \right]^2. \quad (4)$$

其中 $\nabla I = \left(\frac{\partial I}{\partial x}, \frac{\partial I}{\partial y}, \frac{\partial I}{\partial \mathbf{p}} \right) = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$.

将式(4)对 $\Delta\mathbf{p}$ 求导数,得到

$$2 \sum_{\mathbf{x} \in \mathcal{A}} \left[\nabla I \frac{\partial \mathbf{W}}{\partial \mathbf{p}} \right]^T \left[I(\mathbf{W}(\mathbf{x}; \mathbf{p})) + \nabla I \frac{\partial \mathbf{W}}{\partial \mathbf{p}} \Delta\mathbf{p} - T(\mathbf{x}) \right]. \quad (5)$$

令式(5)等于零,得出 $\Delta\mathbf{p}$ 的表达式:

$$\Delta\mathbf{p} = \mathbf{H}^{-1} \sum_{\mathbf{x} \in \mathcal{A}} \left[\nabla I \frac{\partial \mathbf{W}}{\partial \mathbf{p}} \right]^T \left[T(\mathbf{x}) - I(\mathbf{W}(\mathbf{x}; \mathbf{p})) \right]. \quad (6)$$

其中 \mathbf{H} 是 Hessian 矩阵,表达式为

$$\mathbf{H} = \sum_{\mathbf{x} \in \mathcal{A}} \left[\nabla I \frac{\partial \mathbf{W}}{\partial \mathbf{p}} \right]^T \left[\nabla I \frac{\partial \mathbf{W}}{\partial \mathbf{p}} \right]. \quad (7)$$

因此根据式(6)计算 $\Delta\mathbf{p}$, 然后根据式(3)更新 \mathbf{p} , 不断迭代,直到 $\Delta\mathbf{p}$ 小于 1 个给定的数值时,就认为已经收敛,最后得到像素的运动 \mathbf{p} . 单层图像光流容易出现局部极小值,为了解决这个问题,引入了图像金字塔.

图像金字塔是指对同 1 个图像进行缩放,得到不同分辨率下的图像,如图 1 所示.以原始的图像作为金字塔的底层,每上升一层,就对下层图像进行一定倍率的缩放,从而得到 1 个金字塔.在计算光流时,先从底层的图像开始计算,然后把上一层的追踪结果作为下一层的光流初始值,不断迭代计算.当原始图像的像素运动较大时,对于缩放到顶层的金字塔,运动就被限制在一个很小的范围内.比如在原始图像中,特征点运动了 40 个像素,那么按照 0.5 的缩放倍率,在上一层的图像中,特征点就只运动了 20 个像素,再上一层,特征点就只运动了 10 个像素.这样就可以避免在优化过程中出现局部最小值导致优化失败.

1.3 VINS 的后端

在完成初始化模块后,就开始 VIO 模块.如图 2 所示,基于滑窗进行紧耦合单目 VIO 模块.

滑动窗口内所有的状态向量被定义为

$$\left. \begin{aligned} \boldsymbol{\chi} &= [\mathbf{x}_0, \mathbf{x}_1, \dots, \mathbf{x}_n, \mathbf{x}_c^b, \lambda_1, \dots, \lambda_m], \\ \mathbf{x}_k &= [\mathbf{p}_{b_k}^w, \mathbf{v}_{b_k}^w, \mathbf{q}_{b_k}^w, \mathbf{b}_a, \mathbf{b}_g], k \in [0, n], \\ \mathbf{x}_c^b &= [\mathbf{p}_c^b, \mathbf{q}_c^b]. \end{aligned} \right\} \quad (8)$$

式中:世界坐标系下第 k 帧的 IMU 状态包含平移

$\mathbf{p}_{b_k}^w$ 、旋转 $\mathbf{q}_{b_k}^w$ 、速度 $\mathbf{v}_{b_k}^w$; IMU 的加速度零偏 \mathbf{b}_a , 角速度零偏 \mathbf{b}_g ; λ_i 表示 3D 特征点的逆深度;上标 w 表示参考系为世界坐标系; \mathbf{x}_c^b 表示由相机坐标系转换到惯性坐标系的变换.

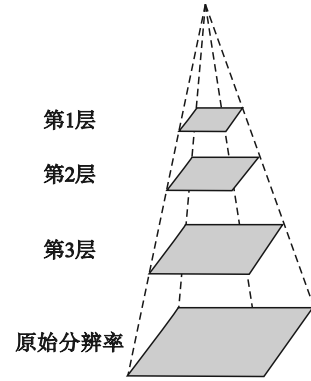


图 1 图像金字塔和多层光流

Fig. 1 Image pyramid and multilayer optical flow

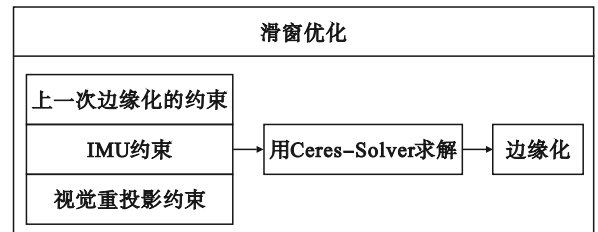


图 2 VINS 后端系统

Fig. 2 VINS back-end system

通过最小二乘优化方法,将滑动窗口内得到的边缘化先验信息、视觉残差以及惯性残差的平方和最小化,目标函数如下:

$$\min \left\{ \left\| \mathbf{r}_p - \mathbf{H}_p \boldsymbol{\chi} \right\|^2 + \sum_{k \in B} \left\| \mathbf{r}_B(\hat{\mathbf{z}}_{b_{k+1}}^w, \boldsymbol{\chi}) \right\|_{\mathbf{P}_{b_{k+1}}^w}^2 + \sum_{(i,j) \in C} \rho \left(\left\| \mathbf{r}_C(\hat{\mathbf{z}}_i^c, \boldsymbol{\chi}) \right\|_{\mathbf{P}_i^c} \right) \right\}. \quad (9)$$

式中: $\mathbf{P}_{b_{k+1}}^w$ 为 IMU 预积分噪声项的协方差矩阵; \mathbf{P}_i^c 为视觉观测噪声的协方差矩阵; Huber 损失函数为

$$\rho(s) = \begin{cases} 1, & s \geq 1, \\ 2\sqrt{s} - 1, & s < 1; \end{cases} \quad (10)$$

$\mathbf{r}_B(\hat{\mathbf{z}}_{b_{k+1}}^w, \boldsymbol{\chi})$ 和 $\mathbf{r}_C(\hat{\mathbf{z}}_i^c, \boldsymbol{\chi})$ 分别是 IMU 和视觉测量值的残差, B 是 IMU 测量值的集合, C 是当前滑窗内至少被观测到 2 次的特征点的集合; $\left\| \mathbf{r}_p - \mathbf{H}_p \boldsymbol{\chi} \right\|^2$ 是从边缘化得到的先验信息.用 Ceres-Solver 求解器来解这个非线性问题.

1.4 边缘化方法

为了约束基于优化的 VIO 的计算复杂度,引

入了边缘化.从滑动窗口中有选择地边缘化 IMU 状态 \mathbf{x}_k 和逆深度 λ_i , 然后边缘化对应的测量值被转换为先验值.

若倒数第 2 帧是关键帧, 就将它留在滑窗, 然后边缘化掉最旧帧对应的测量值. 相反, 如果倒数第 2 帧不是关键帧, 就去掉其视觉测量值, 保留与之相关的 IMU 测量值. 为了保证系统的稀疏性, VINS 没有边缘化所有非关键帧的测量值, 边缘化的策略是保持滑窗中空间上相分离的关键帧, 这可以确保在进行特征点三角化时有充足的视差, 以及在较大的激励下保持加速度测量值的可能性. 边缘化是用 Schur 消元来完成的, 根据所有被去掉的状态量的边缘化测量值来构造新的先验值, 再把它加入到现有的先验中.

边缘化的方法为 Schur 消元, 设增量方程为

$$\mathbf{H}\delta\mathbf{x}=\mathbf{g}. \quad (11)$$

其中: $\mathbf{H}=\mathbf{J}^T\mathbf{J}$; $\mathbf{g}=\mathbf{J}^T\mathbf{e}$, \mathbf{e} 是残差, \mathbf{J} 是残差对状态量的雅可比矩阵.

由于引入滑窗后, 对于滑到窗口之外的帧不再优化其参数, 但其与滑窗内的数据仍然存在约束, 直接丢掉这些窗外帧会造成约束信息丢失, 所以要将其封装成先验信息. 假设 $\delta\mathbf{x}$ 中待边缘化的状态量是 $\delta\mathbf{x}_a$, 留在滑窗中的状态量是 $\delta\mathbf{x}_b$, 则上述增量方程可变为

$$\begin{bmatrix} \mathbf{A}_a & \mathbf{A}_b \\ \mathbf{A}_b^T & \mathbf{A}_c \end{bmatrix} \begin{bmatrix} \delta\mathbf{x}_a \\ \delta\mathbf{x}_b \end{bmatrix} = \begin{bmatrix} \mathbf{g}_a \\ \mathbf{g}_b \end{bmatrix}. \quad (12)$$

方程同时左乘矩阵

$$\begin{bmatrix} \mathbf{I} & \mathbf{0} \\ -\mathbf{A}_b^T\mathbf{A}_a^{-1} & \mathbf{I} \end{bmatrix}, \quad (13)$$

得到如下矩阵:

$$\begin{bmatrix} \mathbf{A}_a & \mathbf{A}_b \\ \mathbf{0} & \mathbf{A}_c - \mathbf{A}_b^T\mathbf{A}_a^{-1}\mathbf{A}_b \end{bmatrix} \begin{bmatrix} \delta\mathbf{x}_a \\ \delta\mathbf{x}_b \end{bmatrix} = \begin{bmatrix} \mathbf{g}_a \\ \mathbf{g}_b - \mathbf{A}_b^T\mathbf{A}_a^{-1}\mathbf{g}_a \end{bmatrix}. \quad (14)$$

将该矩阵按第二行展开, 得到

$$(\mathbf{A}_c - \mathbf{A}_b^T\mathbf{A}_a^{-1}\mathbf{A}_b)\delta\mathbf{x}_b = \mathbf{g}_b - \mathbf{A}_b^T\mathbf{A}_a^{-1}\mathbf{g}_a. \quad (15)$$

将 $\mathbf{A}_c - \mathbf{A}_b^T\mathbf{A}_a^{-1}\mathbf{A}_b$ 记为 \mathbf{H}' , 将 $\mathbf{g}_b - \mathbf{A}_b^T\mathbf{A}_a^{-1}\mathbf{g}_a$ 记为 \mathbf{g}' , 得到新的增量方程:

$$\mathbf{H}'\delta\mathbf{x}=\mathbf{g}'. \quad (16)$$

新的 \mathbf{H}' 与 \mathbf{g}' 也满足如下关系:

$$\left. \begin{aligned} \mathbf{H}' &= \mathbf{J}'^T\mathbf{J}', \\ \mathbf{g}' &= \mathbf{J}'\mathbf{e}'. \end{aligned} \right\} \quad (17)$$

根据此时的 \mathbf{H}' 与 \mathbf{g}' , 求出

$$\left. \begin{aligned} \mathbf{J}' &= \mathbf{S}^{\frac{1}{2}}\mathbf{V}^T, \\ \mathbf{e}' &= -\mathbf{S}^{-\frac{1}{2}}\mathbf{V}^T\mathbf{g}'. \end{aligned} \right\} \quad (18)$$

其中 \mathbf{S} 和 \mathbf{V} 是将 \mathbf{H}' 进行特征值分解得到的矩阵.

然后将新的 \mathbf{J}' 和 \mathbf{e}' 添加到边缘化的约束中, 并封装为先验信息.

2 前端光流法改进策略

在 VINS 的前端部分, 通过光流法提取到特征点后, 进行外点剔除, 并将特征点均匀化. 为了进一步提高提取到的特征点的精度, 在用光流法提取特征点后, 通过 OpenCV 自带的光流法库函数, 采用逆向光流法往回匹配, 再剔除追踪失败的特征点、位于图像边界之外的特征点, 以及剔除离上一个特征点过远的点, 从而增加匹配精度. 改进后的前端流程如图 3 所示.

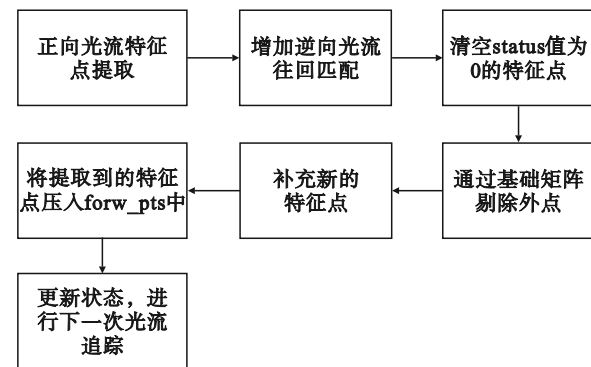


图 3 改进后的前端流程

Fig. 3 Improved front-end process

为了验证本节提出的改进算法的可行性与有效性, 利用 EuRoc 数据集中的 MH_01, MH_02, MH_03, MH_04 四个不同特征数量的数据集进行实验验证, 计算机配置为 Ubuntu18.04, 处理器为 Intel i5-7300HQ, 主频为 2.50 Hz, 运行内存为 8 GB, 利用开源机器人系统 ROS 进行实验验证. 用均方根误差 (root mean square error, RMSE) 作为精度的定性评估指标, 在不同数据集上运行源码与改进算法, 将得到的轨迹文件分别与真实值对比, 得到各自的 RMSE, 如表 1 所示.

表 1 前端的 RMSE 的实验对比

Table 1 Experimental comparison of front-end RMSE

数据集	RMSE		对比结果/%
	源码	改进算法	
MH_01	0.181 302	0.179 712	0.88
MH_02	0.182 778	0.180 029	1.50
MH_03	0.405 599	0.404 163	0.35
MH_04	0.393 492	0.402 433	-2.27

通过表 1 的实验结果对比, 发现改进算法的均方根误差与源码相比, 虽然在数据集 MH_01, MH_02, MH_03 上有所减少, 但减少幅度不大, 最

后得出本节提出的改进方法对精度的提升效果不理想. 主要原因为:对于原本的正向光流法,每进行一次迭代,都需要重新计算 Hessian 矩阵 H , 计算量很大,因此融合了逆向光流法. 在交换模板图像与输入图像的地位后,代价函数变为

$$\sum_x [T(W(x; \Delta p)) - I(W(x; p))]^2. \quad (19)$$

由此得到的新的 Hessian 矩阵 H 与 p 无关,可以提前计算,并反复使用,从而减少了计算量. 但是因为 2 幅图像之间的时间间隔只有 0.3 s, 所以进行光流前后的 2 幅图像并没有太大的差别,采用逆向光流法不能将算法的精度提升太多. 因此,对 VINS 的后端进行改进,并融合边缘化改进策略.

3 边缘化算法改进策略

在滑窗优化的过程中,必然会有新的图像帧进来以及旧的图像帧离开. 边缘化就是为了充分利用需要去掉的图像帧的信息,比如先验信息、IMU 信息等,将它们转化为先验. 主要针对边缘化中残差信息进行优化,为了加速边缘化,首先对除相机位姿以外的部分进行边缘化,然后对相机位姿进行边缘化. 因为与位姿有关的矩阵块比较大,分开进行边缘化会加快速度.

改进算法的伪代码如表 2 所示.

为了验证本文提出的改进算法的可行性与有效性,利用 EuRoc 数据集集中的 MH_01、MH_02、MH_03、MH_04 四个不同特征数量的数据集进行实验验证. 在选定的 4 个数据集上分别运行源码和改进的算法,对于每种算法都分别运行 5 次,每次在输出的边缘化的用时结果中,从最新的结果开始,倒着取 10 次边缘化所用的时间,计算其平均值,再求出这 5 次结果的平均数、中位数、最大值和最小值,以减少偶然条件对定性测

试的影响,结果见表 3. 为了更清晰地显示改进算法较源码的提升情况,在表 4 中展示了提升的比率.

表 2 边缘化改进算法的伪代码
Table 2 Pseudocode for marginal optimization algorithms

步骤	边缘化优化
1	localSize(parameter_block_size[it. first])>4 marg_pose_index_size<-pos,localSize (parameter_block_size[it. first])
2	marg_pose_size<-marg_pose_size+localSize (parameter_block_size[it. first]) //对要边缘化的变量参数根据维度赋予 id
3	m ₁ <-m-marg_pose_size; n ₁ <-n+marg_pose_size;
4	for (auto iter: marg_pose_index_size) //遍历 marg_pose_index_size 的所有元素 swap(A_reorder. block(idx,0,size,pos), A_reorder. block(idx+size,0,m-idx-size,pos)) //将 row i 移到矩阵的最下面 swap(A_reorder. block(0,idx,pos,size),A_reorder. block(0,idx+size,pos,m-idx-size)) //将 col i 移到矩阵的最右面 swap(b_reorder. segment(idx,size),b_reorder. segment(idx+size,m-idx-size)) //将 b i 移到向量的最下面
6	tempA<-Arm1 * Amm_inv1; A<-Arr1-tempA * Amr1; b<-brr1-tempA * bmm1; //用 Schur 消元进行边缘化
7	m ₂ <-m-m ₁ ;n ₂ <-n; MatrixXd tempB=Arm * Amm_inv;
8	A=Arr-tempB * Amr; b=brr-tempB * bmm; //对剩余的部分进行边缘化

表 3 边缘化用时对比
Table 3 Comparison of marginalization time

数据集	源码用时				改进算法用时				ms
	中位数	平均数	最小值	最大值	中位数	平均数	最小值	最大值	
MH_01	11 918. 92	11 877. 99	11 706. 68	11 977. 77	8 550. 97	8 578. 00	8 481. 91	8 677. 89	
MH_02	9 715. 99	9 738. 60	9 567. 80	9 931. 28	7 105. 28	7 059. 44	6 643. 53	7 436. 98	
MH_03	9 405. 12	9 427. 55	9 289. 16	9 649. 91	7 257. 35	7 129. 18	6 726. 81	7 448. 28	
MH_04	6 812. 63	6 904. 28	6 658. 04	7 311. 03	5 227. 38	5 251. 68	5 204. 86	5 318. 54	

表 4 较源码提升的比率

Table 4 Percentage improvement over source code %

数据集	提升比率			
	中位数	平均数	最小值	最大值
MH_01	28.26	27.78	27.55	27.55
MH_02	26.87	27.51	30.56	25.12
MH_03	22.84	24.38	27.58	22.82
MH_04	23.27	23.94	21.83	27.25

根据表 3 和表 4 可以看出,在 4 个数据集上,与源码相比,改进算法加快了边缘化的速度,在输出的边缘化时间中,平均数、中位数、最大值和最小值都得到较大幅度的减小,说明本节改进的边缘化优化算法加快了边缘化的效率.在不同数据集上分别运行 VINS 源码与改进算法,用 Evo (evaluation of odometry) 对比得出轨迹文件,如图

4 所示,图中右侧的数据表示绝对轨迹误差,是数据集轨迹的位姿真实值,与 SLAM 系统对应位置的位姿估计值之间的差值,常用该标准来评价 SLAM 系统的性能.通过对图 4 进行分析可知,源码与改进算法在 4 个数据集上的轨迹相差不大,说明边缘化改进算法对轨迹精度的影响不是很大.为了对改进的算法精度作进一步的定量比较,本节使用均方根误差(RMSE)作为定量评估指标.对于源码和修改算法,都会在相应的数据集上运行 5 次,并将输出的轨迹文件与真实数据集的轨迹进行对齐,以获得 RMSE 的平均数、中位数、最大值和最小值,来减少偶然条件对定量测试的影响,结果见表 5.同时在表 6 中展现了与源码相差的百分比.

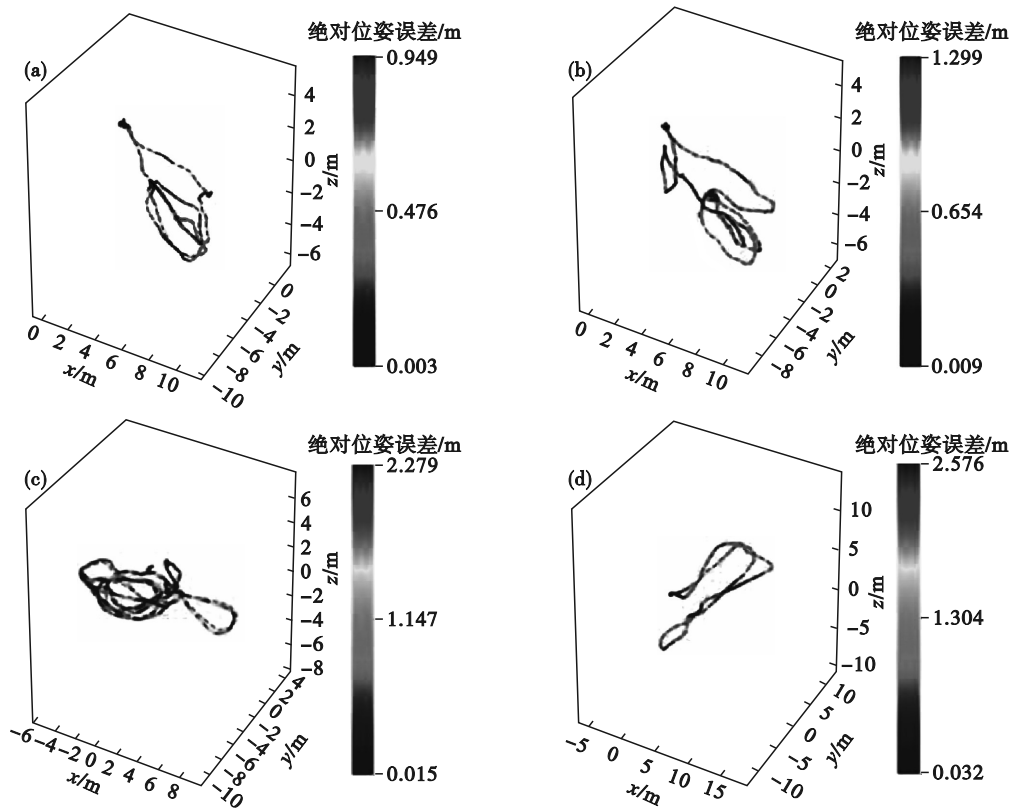


图 4 源码与改进算法在不同数据集上的轨迹对比

Fig. 4 Trajectory comparison of source code and improved algorithm in different data sets

(a)—MH_01; (b)—MH_02; (c)—MH_03; (d)—MH_04.

表 5 后端的 RMSE 实验对比

Table 5 Experimental comparison of back-end RMSE

数据集	源码的 RMSE				改进算法的 RMSE			
	中位数	平均数	最小值	最大值	中位数	平均数	最小值	最大值
MH_01	0.183	0.183	0.182	0.183	0.177	0.177	0.176	0.177
MH_02	0.182	0.183	0.182	0.183	0.180	0.181	0.179	0.184
MH_03	0.406	0.406	0.403	0.410	0.406	0.407	0.406	0.410
MH_04	0.393	0.397	0.392	0.414	0.394	0.395	0.394	0.396

表6 与源码相差的百分比

Table 6 Percentage difference from source code %

数据集	相差百分比			
	中位数	平均数	最小值	最大值
MH_01	3.28	3.28	3.30	3.28
MH_02	1.10	1.09	1.65	-0.55
MH_03	0.00	-0.25	-0.74	0.00
MH_04	-0.25	0.65	-0.51	4.35

通过表5和表6可以看出,在4个数据集的实验结果中,改进算法与源码的均方根误差都非常接近,这说明本文改进后的边缘化优化算法对轨迹精度的影响可控。

4 结 论

1) 在 VINS 的前端,增加了逆向光流法,意图增加算法的精度,但经过实验验证,效果不明显,并分析了效果不明显的原因。

2) 由于对 VINS 前端的改进效果不明显,所以没有进行前后端的融合,而是单独改进了后端的边缘化部分.在 VINS 的后端融合了一种边缘化改进算法,通过将边缘化的过程分为两步,增加了边缘化的速度,减少了程序运行的时间,提高了算法的实时性。

3) 为了验证改进的后端算法的有效性,在公开的 EuRoc 数据集上进行了实验,并与 VINS-Mono 的源码进行了对比分析.实验结果表明,改进算法的边缘化运行时间比源码平均减少了 25.9%,且改进算法对精度的影响可控。

参考文献:

- [1] Davison A J, Reid I D, Molton N D, et al. MonoSLAM: real-time single camera SLAM [J]. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2007, 29 (6): 1052-1067.
- [2] Klein G, Murray D. Parallel tracking and mapping for small AR workspaces [C]//IEEE and ACM International Symposium on Mixed and Augmented Reality. Nara, 2007: 225-234.
- [3] Engel J, Schöps T, Cremers D. LSD-SLAM: large-scale direct monocular SLAM [C]//European Conference on Computer Vision. Cham: Springer, 2014: 834-849.
- [4] Engel J, Sturm J, Cremers D. Semi-dense visual odometry for a monocular camera [C]//IEEE International Conference on Computer Vision. Sydney, 2013: 1449-1456.
- [5] Forster C, Pizzoli M, Scaramuzza D. SVO: fast semi-direct monocular visual odometry [C]//IEEE International Conference on Robotics and Automation. Hong Kong, 2014: 15-22.
- [6] Newcombe R A, Lovegrove S J, Davison A J. DTAM: dense tracking and mapping in real-time [C]//2011 International Conference on Computer Vision. Barcelona, 2011: 2320-2327.
- [7] Engel J, Koltun V, Cremers D. Direct sparse odometry [J]. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2018, 40 (3): 611-625.
- [8] Kerl C, Sturm J, Cremers D. Dense visual SLAM for RGB-D cameras [C]//IEEE/RSS International Conference on Intelligent Robots and Systems. Tokyo, 2013: 2100-2106.
- [9] Mur-Artal R, Montiel J M, Tardós J D. ORB-SLAM: a versatile and accurate monocular SLAM system [J]. *IEEE Transactions on Robotics*, 2015, 31 (5): 1147-1163.
- [10] Mourikis A I, Roumeliotis S I. A multi-state constraint Kalman filter for vision-aided inertial navigation [C]//IEEE International Conference on Robotics and Automation. Rome, 2007: 3565-3572.
- [11] Bloesch M, Omari S, Hutter M, et al. Robust visual inertial odometry using a direct EKF-based approach [C]//IEEE/RSS International Conference on Intelligent Robots and Systems (IROS). Hamburg, 2015: 298-304.
- [12] Lynen S, Achtelik M W, Weiss S, et al. A robust and modular multi-sensor fusion approach applied to MAV navigation [C]//IEEE/RSS International Conference on Intelligent Robots and Systems. Tokyo, 2013: 3923-3929.
- [13] Leutenegger S, Lynen S, Bosse M, et al. Keyframe-based visual-inertial odometry using nonlinear optimization [J]. *The International Journal of Robotics Research*, 2015, 34 (3): 314-334.
- [14] Qin T, Li P L, Shen S J. VINS-Mono: a robust and versatile monocular visual-inertial state estimator [J]. *IEEE Transactions on Robotics*, 2018, 34 (4): 1004-1020.
- [15] Martinelli A. Closed-form solution of visual-inertial structure from motion [J]. *International Journal of Computer Vision*, 2014, 106 (2): 138-152.
- [16] Lucas B D, Kanade T. An iterative image registration technique with an application to stereo vision [C]//International Joint Conference on Artificial Intelligence. Vancouver, 1981: 674-679.