

一种结合光流追踪的双目视觉里程计算法

李伟东, 朱旭浩

(大连理工大学 汽车工程学院, 辽宁 大连 116024)

摘要: 针对特征点法视觉里程计频繁计算和匹配描述子导致系统实时性能变差的问题, 提出一种结合光流追踪的双目视觉里程计算法; 首先进行初始化, 生成初始的关键帧和地图点, 随后在追踪线程中使用光流追踪特征点获取匹配关系, 计算并优化相机位姿; 满足生成关键帧的条件后, 将当前帧设置为关键帧, 提取图像的 ORB 特征点, 并使用描述子匹配获取与上一关键帧特征点的匹配关系, 三角化生成新的地图点; 最后在优化线程中将新的关键帧和地图点使用滑动窗口算法进行优化, 剔除冗余关键帧和地图点; 在 KITTI 数据集上的实验结果表明, 所提出的算法轨迹误差与双目模式下的 ORB-SLAM3 算法处于同一水平, 同时实时性能有大幅度提高, 追踪每一帧图像的平均时间从 52 ms 降至 16 ms, 在保证高精度的情况下运行速度大大提高, 具有较高的实用价值。

关键词: 视觉里程计; 同时定位与建图; 特征匹配; 实时性能; 轨迹误差

Stereo Visual Odometry Algorithm Combined with Optical Flow Tracking

LI Weidong, ZHU Xuhao

(School of Automotive Engineering, Dalian University of Technology, Dalian 116024, China)

Abstract: Aiming at the problem that the real-time performance of the system becomes low due to the frequent calculation and matching descriptors of the feature-based visual odometry, a stereo visual odometry combined with optical flow tracking is proposed. Firstly, the initialization is performed to generate the initial key frames and map points, then in the tracking thread, the optical flow tracking feature points are used to obtain the matching relationship, calculate and optimize the camera pose. After meeting the conditions for the generating key frames, the current frame is taken as a key frame, the ORB feature points of the image are extracted, the descriptor matching is used to obtain the matching relationship with the feature points of the previous key frame, and triangulate to generate new map points. Finally, the new key frames and map points are optimized using the sliding window algorithm in the optimization thread, and the redundant key frames and map points are eliminated. In the KITTI data, the experimental results show that the trajectory error of the proposed algorithm is at the same level as that of the ORB-SLAM3 algorithm in the trajectory err and stereo mode, and the real-time performance is greatly improved. The average time for tracking each frame image is reduced from 52 ms to 16 ms. In the case of ensuring high precision, the running speed is greatly improved, and it has a high practical value.

Keywords: visual odometry; simultaneous localization and mapping; feature matching; real-time performance; trajectory error

0 引言

里程计技术最早出现于机器人领域, 它是指机器人在未知环境中通过传感器捕捉环境特征来确定自身位置和姿态, 同时构建一个增量地图。使用相机作为传感器的里程计称为视觉里程计。相较于激光雷达传感器, 相机部署方便, 价格低廉, 并且可以获取更为丰富的视觉信息。随着自动驾驶技术的发展, 视觉里程计也越来越多地被用于对载具进行定位和地图构建, 因此基于视觉的里程计受到了广泛关注。基于特征匹配的特征点法通常需要提取特征描述子并对其进行匹配, 这个过程会消耗较多的资源和时间, 在一些无法搭载高算力设备的自动驾驶场景中实时性较差;

直接法则不需要耗费时间计算提取描述子, 在只计算稀疏像素点的光度误差时速度较快, 但是对于光照变化非常敏感, 精度低于特征点法视觉里程计, 并且缺乏描述子用于系统追踪失败时进行重定位。因此一个轻量快速的高精度视觉里程计仍然是一个关键的研究问题。

根据数据关联方式的不同, 视觉里程计可以分为基于光度误差的直接法和基于特征匹配的特征点法^[1]。直接法视觉里程计通过最小化不同图像之间的光度误差来恢复相机运动。LSD-SLAM (Large Scale Direct Monocular SLAM)^[2]是 Engel 等人提出的一种直接法视觉 SLAM, 它基于单目相机运行, 并且可以在 CPU 上进行半稠密建图。为了提高全局精度, 系统还加入了回环检测模块。相较于

收稿日期: 2023-03-09; 修回日期: 2023-04-14。

基金项目: 辽宁省“揭榜挂帅”科技计划(重大)项目(ZX20220560)。

作者简介: 李伟东(1975-), 男, 博士, 副教授。

引用格式: 李伟东, 朱旭浩. 一种结合光流追踪的双目视觉里程计算法[J]. 计算机测量与控制, 2024, 32(2): 174-180.

传统直接法, LSD-SLAM 将光度误差以及深度噪声一起进行优化从而实现了较好的效果, 但是由于采用了单目相机, 所以无法恢复绝对尺度信息。Stereo LSD-SLAM^[3]则是将 LSD-SLAM 拓展到双目相机上, 解决了单目相机无法恢复绝对尺度信息的问题。DSO (Direct Sparse Odometry)^[4]是一种纯直接法视觉里程计, 它将数据关联以及位姿估计统一成一个非线性优化问题, 并且使用了滑动窗口算法, 既控制了优化规模, 提高优化速度, 同时又保证了一定的系统精度。其缺点是由于图像之间的光度误差函数是一个非凸函数, 因而优化时非常容易陷入局部最优值, 只适合于移动较小的场景中。SVO (Semi Direct Monocular Visual Odometry)^[5]是 Forster 等提出的一个结合了特征点法和直接法的半直接法视觉里程计。其首先提取稀疏特征点, 使用直接法进行不同图像之间对齐从而计算位姿; 同时使用深度滤波器来估计深度信息。优点是运行速度非常快, 在低功耗设备上也有较高的帧率; 缺点是其对光照情况非常敏感, 在光照变化大的情况下非常容易追踪失败。

特征点法视觉里程计通过提取视觉特征, 获取匹配关系后使用光束法平差^[6]计算和优化相机的位姿与地图点位置。PTAM^[7]是 Klein 等人提出的一个视觉里程计系统, 其包含了追踪和建图两个线程来同时恢复相机运动并优化地图点和位姿, 首次将多线程用于视觉里程计算法中。Raulmur 等提出的 ORB-SLAM^[8]则在 PTAM 两个线程基础上增加了回环检测线程, 其利用词袋模型^[9]进行相似度检测, 在检测到发生回环后进行回环矫正来全局优化。ORB-SLAM 采用的传感器为单目相机, 无法恢复绝对尺度, 并且非常容易出现尺度漂移的现象。在此基础之上 ORB-SLAM2^[10]加入了对双目相机以及 RGBD 相机的支持, 不仅可以恢复绝对尺度, 并且在 RGBD 模式下可以实现稠密建图。随后的 ORB-SLAM3^[11]则加入了鱼眼相机模型, 并且可以融合 IMU 传感器, 精度有了很大的提高。除了点特征外, 线特征也可以用于位姿估计, 在特征点缺失的场景下会有更好的鲁棒性。PL-SLAM^[12]是 Pumarola 等提出的一个将线特征用于视觉里程计上的视觉 SLAM 算法, 该算法结合了线特征和点特征, 在一些特征点缺失的场景中更加鲁棒, 精度更高, 但是同时也提高了计算复杂度, 导致系统计算资源消耗变大。Gomez 等则将 PL-SLAM 扩展到双目相机上, 不仅提高了系统精度, 同时还可以恢复绝对尺度^[13]。

精度与运行速度是评价视觉里程计的两个重要指标。特征点法视觉里程计会消耗大量时间去计算每一帧图像中关键点的描述子, 对实时性有较大影响; 直接法视觉里程计省去了描述子计算过程, 但是对于光照极为敏感, 相较于特征点法精度有所降低, 同时缺乏描述子用于在系统追踪失败时进行重定位。为了实现保持较高精度的同时有更好的实时性能, 提出了一种结合了光流追踪的视觉里程计算法。该算法基于特征点法, 在追踪阶段主要通过光流追

踪来获得匹配关系; 生成关键帧时使用描述子来获取匹配关系, 并且在追踪失败时会使用描述子匹配进行重定位。由于只在提取关键帧和重定位时计算和匹配描述子, 算法的运行速度有了大幅提升, 同时保持了较高的精度。

1 视觉里程计算法

视觉里程计选取的传感器为双目相机。相较于单目相机, 双目相机可以计算得到像素点深度信息, 解决了尺度不确定的问题, 并且在追踪过程中有着更好的鲁棒性。算法包括追踪线程和滑动窗口优化线程两大部分。追踪线程主要用于计算相机当前位姿, 滑动窗口优化线程则会对关键帧及地图点进行优化, 保证较高的计算精度。相机捕捉到的图像作为输入, 输出内容包括关键帧和地图点, 两者共同构成了地图。每个关键帧也保留了以其为参考的普通帧相对位姿。

视觉里程计在运行时首先会进行初始化操作, 通过提取当前图像中的 ORB 特征点进行匹配。由于初始化生成的地图和位姿对整个系统运行的精度有着较大影响, 因此初始化过程所提取特征点数量较多。当匹配成功的特征点对数量满足要求后会三角化生成新的地图点, 并将对应的图像设置为关键帧。在生成初始地图后系统会对其中的关键帧和地图点位姿进行一次全体优化, 保证系统的初始精度, 同时会根据关键帧的位姿来计算并初始化运动模型。当成功完成初始化工作后, 系统会使用运动模型进行追踪, 若系统追踪失败则会切换到参考关键帧追踪进行重定位来恢复相机当前的位姿和运动模型。在运行过程中系统会判断当前是否需要生成新的关键帧。当满足生成关键帧的条件时, 系统会对当前帧进行 ORB 特征点提取操作, 同时将其与上一关键帧中所提取的 ORB 特征点进行匹配, 并通过三角化生成新的地图点。同时将新的关键帧和地图点插入到滑动窗口中进行优化, 剔除冗余关键帧和地图点。该算法框架如图 1 所示。

1.1 特征提取

常见的视觉特征点有 SIFT 特征点、SURF 特征点以及 ORB 特征点。相较于 SIFT 和 SURF 特征点, ORB 特征点在保持旋转、尺度不变性的前提下有更高的计算速度, 因此系统选取 ORB 特征点来进行数据关联。

ORB^[14]特征点包含具有方向信息的 FAST 角点和 BRIEF 描述子两部分。FAST 角点通过检测像素灰度变化来确定, 其原理如图 2 所示。FAST 角点通过以下几个步骤进行提取:

- 1) 选取一个像素 p , 记录它的亮度 I ;
- 2) 设定阈值 T , 并以像素 p 为中心, 选取半径为 3 的圆周上的像素点;
- 3) 若有连续 N 个点的亮度大于 $I+T$ 或者小于 $I-T$, 则将像素点 p 选取为 FAST 角点。

传统的提取方法阈值亮度阈值 T 固定不变, 容易导致在整张图像中出现 FAST 角点过于集中和分布不均匀的现

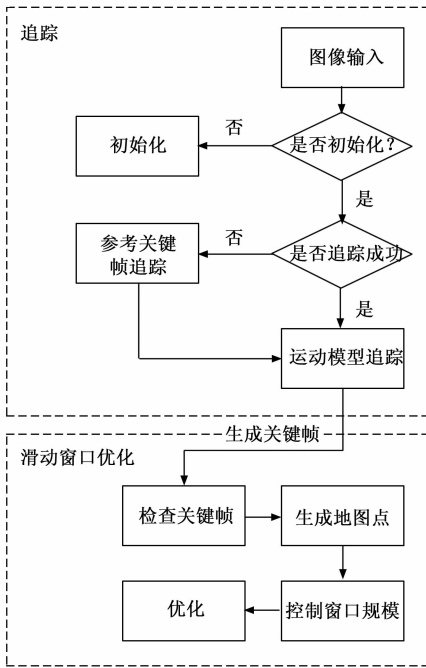


图 1 算法框架

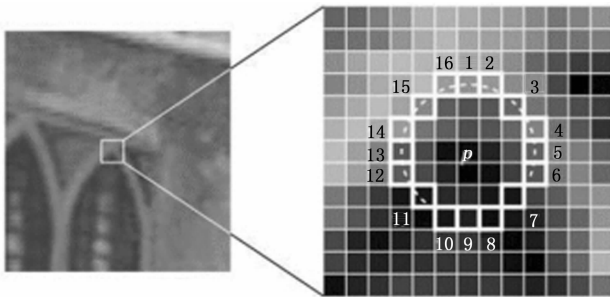


图 2 FAST 角点

象,使得系统计算精度降低。为了缓解该现象,系统在提取 FAST 角点时会把图像分割成多个区块,对每个区块进行 FAST 角点提取。若某个区域提取到的 FAST 角点数量过少,系统会将该区域的亮度阈值 T 减小,从而提取到更多的角点^[15]。当 FAST 角点提取完毕后,系统会使用四叉树均分策略剔除掉冗余的角点,使分布更加均匀。

FAST 角点方向通过灰度质心法来确定,步骤如下:

1) 在一个图像块 B 中定义它的矩为:

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

其中: $p, q = \{0, 1\}$ 。

2) 图像块的质心 C 为:

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

3) 设 O 为图像块的几何中心,通过向量将 FAST 角点方向定义为:

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

通过灰度质心法计算得到 FAST 角点方向可以保证特

征点的旋转不变性。

BRIEF^[16] 描述子是一种二进制描述子,通过比较 FAST 角点附近的像素亮度关系可以获得一串由 0 和 1 构成的向量。由于其使用了二进制表达,因此相较于其他描述子有着非常快的计算速度。旋转不变性通过 FAST 角点的方向来确定,尺度不变性则引入图像金字塔来实现。如图 3 所示,通过将图像按照一定比例缩放形成图像金字塔,在每一层进行特征点提取,从而保证在特征匹配时的尺度不变性。

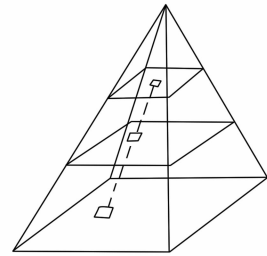


图 3 图像金字塔

1.2 特征匹配

特征匹配用于获取同一特征点在不同图像中的像素位置。获取特征点匹配关系后,系统会通过最小化重投影误差来得到相机位姿。为了提高系统的运行速度,同时保证计算精度,系统采用了光流追踪和描述子匹配两种匹配方法。

运动模型追踪通过光流追踪特征点来获得匹配关系。光流用于描述图像之间像素的运动,通过光流可以追踪特征点在不同帧之间的位置从而获取匹配关系。相较于描述子匹配,光流追踪可以省去提取描述子的过程,速度更快。算法采用 Lucas-Kanade^[17] 光流,简称 LK 光流,其工作原理如图 4 所示。LK 光流是一种稀疏光流,用于追踪稀疏像素,其基于两个假设:

1) 灰度不变假设,即空间中任意一点的灰度值在不同图像中是相同的。图像中 (x, y) 处的像素从 t 时刻运动到 $t + dt$ 时刻,坐标变为 $(x + dx, y + dy)$,其灰度值不变,用公式表示为 $I(x + dx, y + dy, t + dt) = I(x, y, t)$ 。对其进行泰勒展开,整理可得:

$$\begin{bmatrix} I_x & I_y \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = -I_t \quad (4)$$

其中: $I_x = \frac{\partial I}{\partial x}$, $I_y = \frac{\partial I}{\partial y}$, $u = \frac{dx}{dt}$, $v = \frac{dy}{dt}$ 。

2) 在同一个窗口内的所有像素会做相同的运动。假设在同一窗口中选取了 n 个像素点,那么则有:

$$\begin{bmatrix} I_x & I_y \end{bmatrix}_k \begin{bmatrix} u \\ v \end{bmatrix} = -I_{tk} \quad k = 1, \dots, n \quad (5)$$

当相机快速移动时,光流追踪特征点容易失败。为了缓解该现象,系统使用了图像金字塔来进行光流追踪。图像金字塔首先对图像进行由精到粗进行降采样处理,同时

上层的光流追踪结果作为下一层的计算初始值, 实现由粗至精追踪光流, 从而缓解快速运动带来的影响。

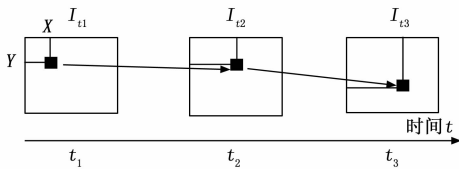


图 4 LK 光流法示意图

通过光流追踪获取特征点会出现一些误匹配现象, 导致系统计算精度降低, 因此系统会进一步使用随机抽样一致性 (RANSAC, random sample consensus)^[18] 算法进行过滤, 剔除错误的匹配关系。通过光流追踪来获取特征点匹配关系不需要计算和匹配描述子, 因此运行速度有了较大提升。

生成关键帧和参考关键帧追踪会使用描述子匹配来获取特征点匹配关系。描述子用于描述关键点周围的信息, 通过对比特征点的描述子可以判断匹配关系。系统采用的描述子为 BRIEF 描述子。它是一种二进制的描述子, 通过计其汉明距离可以确定特征点之间的匹配关系。传统的匹配方法在遇到大量特征点时耗时非常大, 并且会出现大量误匹配现象, 降低系统的实时性能和计算精度。因此系统采用改进后的匹配方法, 通过以下步骤进行特征匹配:

1) 通过运动模型提供的初始位姿计算得出特征点在下一帧图像中的粗略位置, 并与位于该位置半径 8 个像素以内提取到的特征点进行匹配, 选出汉明距离最短且满足阈值要求的特征点作为匹配点。若未成功匹配, 系统则会将其半径提高至 14 个像素, 进一步匹配。

2) 根据灰度质心法计算出的特征点方向来计算其在两帧之间的旋转角度, 并将所有匹配成功的特征点旋转角度进行统计, 将旋转角度偏离较大的特征点剔除, 最后使用 RANSAC 算法进一步剔除外点, 保证系统的整体精度。

1.3 关键帧和地图点

关键帧^[19]是指一系列连续图像中最有代表性的一帧, 其包含的信息质量很高, 相当于普通帧进行过滤和优化的结果。采取关键帧策略可以有效降低信息的冗余度, 控制系统优化的规模, 防止无效或错误信息对系统的干扰。系统会提取关键帧的 ORB 特征点, 并通过提取到的特征点三角化生成地图点, 将其插入到滑动窗口中进行优化。为了确保系统的稳定性, 降低追踪失败的机率, 关键帧的选取采用比较宽松的策略。系统初始化时会将第一帧设置为关键帧, 三角化生成初始地图点, 并将其位姿设置为坐标系原点。初始化后系统在运行过程中若满足以下条件之一则会生成新的关键帧:

- 1) 追踪到的特征点数量小于设定阈值;
- 2) 距离上次生成关键帧超过 10 帧;
- 3) 当前图像与上次生成的关键帧旋转角度或者平移距

离超过设定阈值。

关键帧之间提取并成功匹配的 ORB 特征点会用于三角化生成地图点。由于只采用提取到的 ORB 特征点来三角化生成地图点而非所有的像素点, 因此系统生成的是一个稀疏地图。相较于稠密地图, 建立稀疏地图可以大大降低资源消耗, 提高运行速度, 使视觉里程计可以运行在低算力平台上, 同时满足较高精度的定位需求。

1.4 运动模型追踪与参考关键帧追踪

系统追踪过程中主要采取运动模型追踪, 其通过光流追踪来获取特征点匹配关系, 不需计算和匹配描述子, 因此速度较快。运动模型会根据式 (6) 使用上一帧的初始位姿 T_1 和速度 δT 来计算当前帧的初始位姿 T_2 , 并以 T_2 作为初始值将地图点投影的位置与相机观测到的位置做差, 构建出一个非线性优化问题, 并使用列文伯格—马夸尔特算法进行优化求解。如式 (7) 所示, 其中 K 为相机内参, μ 为特征点的像素坐标, P 为地图点坐标, T 为相机位姿。

$$T_2 = \delta T * T_1 \quad (6)$$

$$T^* = \arg \min_T \frac{1}{2} \sum_{i=1}^n \left\| \mu_i - \frac{1}{s_i} K T P_i \right\|_2^2 \quad (7)$$

系统初始化或者追踪失败时会采取关键帧追踪, 其通过描述子来获取匹配关系。当系统初始时运动模型为空, 此时会进行关键帧生成步骤, 通过描述子匹配来计算相机位姿, 初始化运动模型, 随后使用运动模型追踪。当运动模型追踪失败时, 系统会提取当前帧的 ORB 特征点, 并与滑动窗口内的关键帧进行匹配, 成功匹配后会生成新的关键帧并恢复运动模型, 继续使用运动模型追踪。

1.5 滑动窗口优化

系统生成关键帧后, 系统在后端会对关键帧和地图点进行优化, 其本质是一个非线性优化问题。设 z_{ij} 为相机在 T_i 处观察到的地图点 p_j 产生的数据, 那么观测误差为:

$$e_{ij} = z_{ij} - h(T_i, p_j) \quad (8)$$

系统所优化的整体代价函数为:

$$\frac{1}{2} \sum_{i=1}^m \sum_{j=1}^n \rho_{ij} \Omega_{ij}^{-1} e_{ij} \quad (9)$$

其中: ρ 为鲁棒核函数, h 为相机投影公式, Ω_{ij}^{-1} 为信息矩阵, 与特征点尺度有关。

为了保证系统的运行速度, 系统采用滑动窗口算法^[20]来进行优化。滑动窗口算法是指当系统生成新的关键帧后, 为了维持一定的计算量, 防止优化规模不断扩大, 会设置一个窗口大小, 将新的关键帧插入, 同时将冗余的关键帧移除。若滑动窗口内关键帧数量没有达到阈值, 系统会对窗口内所有关键帧和地图点进行优化。当关键帧数量超过设定的阈值时, 系统会对窗口内的关键帧判断, 将共视程度最小的关键帧设置为冗余关键帧, 并使用舒尔补将其从滑动窗口中边缘化; 对于没有追踪到的地图点则直接将其从滑动窗口中剔除。相较于直接将关键帧位姿固定不进行迭代优化或者直接将关键帧约束信息删除的做法, 该方法可以有效保留冗余关键帧的约束关系, 在保证有较高的计

算精度的同时也控制了用于优化的地图规模，提高了系统的实时性能。

2 实验结果与分析

实验首先会验证特征点提取、光流追踪特征点以及描述子匹配特征点的效果，进行定性分析。最后在数据集上对整个视觉里程计算法进行实验，作为对比，会同时对双目模式下的 ORB-SLAM3 算法进行实验，评估两者轨迹误差以及实时性能，并对实验结果进行分析。操作系统采用的是 Ubuntu18，CPU 型号为 AMD Ryzen 7 4800U，主频 1.80 GHz，内存大小为 16 GB，不使用 GPU 加速，特征点提取数量预设 1 000。数据集采用的是 KITTI^[21] 数据集 00~10 序列，其包含了城市道路、高速公路等多种场景，灰度图由双目相机采集并进行了矫正，同时提供了位姿真实数据。

2.1 特征点提取实验

系统会在初始化过程和生成关键帧时会提取图像中的 ORB 特征点。传统的 ORB 特征点提取方法会出现特征点过于集中的现象，导致计算数值上精度下降。系统所采用的优化后的提取方法可以使 ORB 特征点均匀分布在图像中，并且可以将冗余的特征点剔除，避免出现特征点过于集中的现象，从而提升系统高数值计算精度和稳定性。

在 KITTI 数据集上分别使用传统 ORB 特征点提取方法和改进后的方法进行特征点提取，并将结果输出为图片进行对比。实验的结果如图 5 所示。



图 5 特征点提取实验

从实验结果可以看出，相较于传统的 ORB 特征点提取方法，改进后的方法可以有效地将特征点均匀分布在图像中，避免了特征点集中的现象。

2.2 光流追踪特征点实验

追踪线程主要采用光流追踪特征点获取匹配关系，同时以上一帧图像的位姿为初始值，通过最小化重投影误差来计算和优化得到当前位姿。光流追踪可以避免描述子的计算和匹配过程，使得系统有着较高的运行速度。算法采用的是多层光流，相较于单层光流，在高速运动时仍能较好地追踪特征点。追踪成功后系统会使用 RANSAC 算法过滤掉一些错误的追踪信息，提高系统的精度。在 KITTI 数

据集上，分别使用单层光流和系统所采用的多层光流对特征点进行追踪，并将追踪结果绘制输出进行对比。实验结果如图 6 所示。

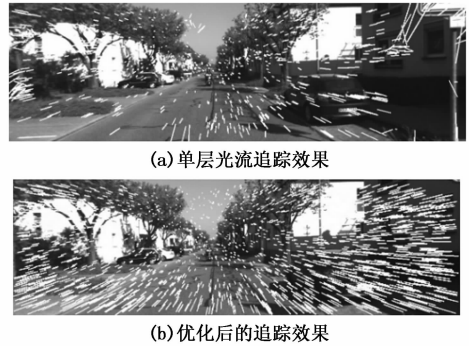


图 6 光流追踪实验

从实验结果可以看出，优化后的多层光流可以追踪到更多的特征点，同时错误的结果也大大减少。

2.3 描述子匹配特征点实验

生成关键帧时系统会提取当前图像中的 ORB 特征点，并使用描述子进行特征匹配。使用描述子汉明距离的传统匹配方法会出现大量错误匹配，在遇到大量特征点时还会严重消耗资源，导致最后的运行速度和计算精度降低。系统采取的优化后的匹配方法，可以有效降低误匹配现象，有着更高的计算精度。在 KITTI 数据集上首先使用恢复运动模型，用于为粗匹配提供初始位姿。随后进行特征匹配，将最终匹配结果和传统方法匹配结果输出，如图 7 所示。

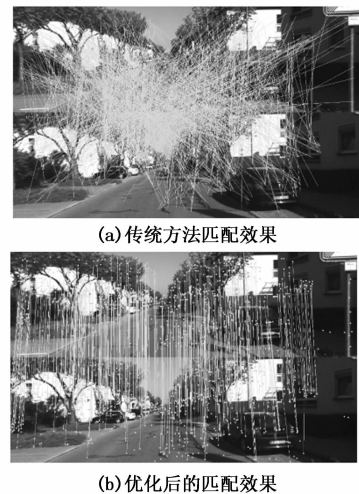


图 7 描述子匹配实验

实验结果表明，相比于传统的匹配方法，改进后的方法可以大大降低误匹配现象，从而保证了系统的精度。

2.4 视觉里程计整体性能实验

评价视觉里程计主要包括轨迹误差以及实时性能两个指标。轨迹误差指标采用绝对平移误差来评估，其计算公式如式 (10) 所示：

$$ATE_{trans} = \sqrt{\frac{1}{N} \sum_{i=1}^N \| trans(T_{gt,i}^{-1} T_{esti,i}) \|_2^2} \quad (10)$$

实时性能则通过计算平均追踪每一帧图像的耗时间来评定。

在 KITTI 数据集 00~10 序列上, 分别测试所提视觉里程计算法和双目模式下的 ORB-SLAM3 算法。由于所提视觉里程计算法没有回环检测模块, 因此在精度评估时还会测试双目模式下去掉回环检测功能的 ORB-SLAM3 算法, 并对结果进行对比分析。每次测试前均会将电脑进行重启操作, 避免每次运行实验结束后对接下来的测试造成影响。运行时系统会记录追踪每一帧所消耗的时间; 运行结束后保存生成的数据集位姿数据, 通过与真实位姿对比计算绝对平移误差, 同时计算追踪每一帧的平均耗时。计算绝对平移误差使用的工具为 evo, 它是一款用于视觉 SLAM 和视觉里程计轨迹评估的工具, 可以评估轨迹与真值之间的误差, 并且支持多种数据集轨迹格式之间的相互转换。计算追踪每一帧所消耗的时间则由系统内置函数实现。为了保证结果的准确性, 每个算法均测试 5 次, 取最后的平均值作为结果进行分析。

图 8 显示了所提视觉里程计算法在 KITTI 数据集各序列上建立的稀疏地图。相较于稠密地图, 稀疏地图消耗的资源更少, 可以仅使用 CPU 而不借助 GPU 加速即可生成和维护, 并且可以满足定位需求。

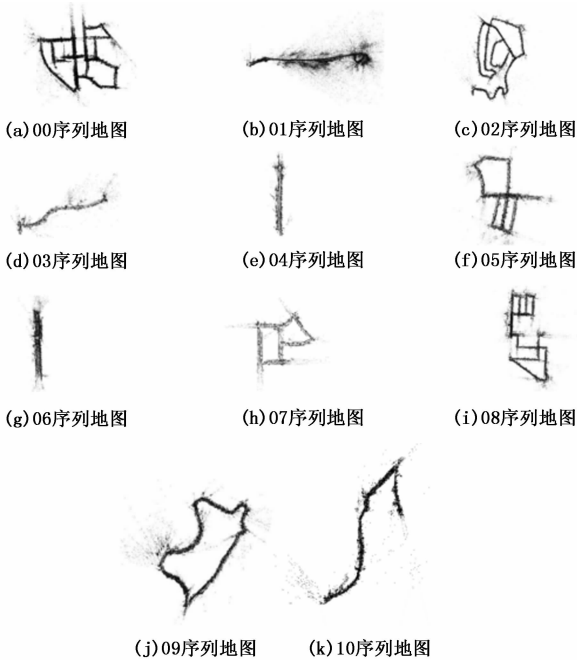


图 8 视觉里程计在 KITTI 数据集 00~10 序列上建立的稀疏地图

表 1 显示了在 KITTI 数据集上的轨迹精度。从表中数据可以看出, 在 00、02、05~09 等序列上发生了回环场景, 因此有回环检测模块并且进行了全局优化的 ORB-SLAM3 算法精度最高, 关闭回环检测功能后精度明显下降; 所提出的视觉里程计算法由于在滑动窗口中采取了边缘化关键帧的策略, 保留了被剔除关键帧的约束信息, 因

此在所有序列中轨迹误差相较于关闭回环检测模块的 ORB-SLAM3 算法更低, 拥有着更高的精度水平, 在发生回环场景中的序列精度会略低于有回环检测功能的 ORB-SLAM3 算法。在没有发生回环场景的 03、04、10 等序列上所提出的视觉里程计算法有着非常高的精度, 与具有完整回环检测模块和全局优化的 ORB-SLAM3 算法精度相当。综上所述可以看出, 在所有序列下所提视觉里程计算法精度均高于去掉回环检测功能的 ORB-SLAM3 算法; 在无回环的场景下精度与进行了回环检测和全局优化的 ORB-SLAM3 算法处于同一水平, 仅在大范围有回环场景的情况下精度有所下降, 但仍处于较高水平。

表 1 绝对平移误差对比

序列	绝对平移误差/m		
	ORB-SLAM3(无回环)	ORB-SLAM3(有回环)	所提算法
00	4.53	1.32	3.07
01	10.78	10.24	10.31
02	8.55	6.38	7.92
03	0.69	0.67	0.62
04	0.21	0.24	0.21
05	1.89	0.82	1.21
06	2.53	0.58	1.92
07	1.56	0.73	1.41
08	3.64	3.33	3.59
09	3.25	3.11	3.24
10	1.05	1.03	1.03

表 2 显示了算法在 KITTI 数据集上平均每追踪一帧图像所需的时间。由于所提算法在追踪线程主要通过光流追踪特征点来获取匹配关系, 省去了重复计算描述子的过程, 因此运行速度有了极大的提升。相比于双目模式下的 ORB-SLAM3 算法, 所提出的视觉里程计算法追踪每一帧所花费的平均时间由 52 ms 降低到 16 ms。其中序列 01 是高速公路场景, 载具处于高速运动状态, 为了保证追踪稳定系统会更频繁地生成关键帧, 因此处理时间相比其他低速场景会

表 2 追踪每一帧图像平均耗时

序列	图像数量	算法平均耗时/ms	
		ORB-SLAM3	所提算法
00	4 541	51	16
01	1 101	61	31
02	4 661	52	14
03	801	50	15
04	271	53	15
05	2 791	52	17
06	1 101	54	16
07	1 102	50	17
08	4 071	51	16
09	1 591	52	14
10	1 201	54	14
总计	23 202	52	16

有所增加,但是仍低于 ORB-SLAM3 算法;而在城市路段中低速场景下,追踪每一帧的耗时仅需不到 20 ms,实时性能有着巨大的提升。综上可以看出,所提视觉里程计算法在实时性能上有着巨大的提高,在低算力设备上有着明显的优势。

3 结束语

针对特征点法视觉里程计频繁计算和匹配描述子导致系统实时性能变差的问题,提出一种结合光流追踪的双目视觉里程计算法。该算法在追踪线程主要通过光流追踪特征点来获取匹配关系,计算并优化相机位姿;生成关键帧时提取和匹配 ORB 特征点,并三角化出新的地图点,随后将新的关键帧和地图点插入滑动窗口中进行优化。由于该算法大部分时间采用光流追踪,不需要计算和匹配描述子,同时采用滑动窗口算法来优化,保留了被剔除的关键帧约束关系,因而有较快的运动速度和较高的精度。在 KITTI 数据集上的实验表明,所提视觉里程计算法可以在较高精度的情况下,有着非常快的运行速度,具有较高的实用价值。由于本文所提出的视觉里程计算法没有加入回环检测和全局优化功能,因此不是一个完整的视觉 SLAM 系统,在有回环的场景下精度有所降低。因此加入回环检测和全局优化功能是下一阶段的主要研究内容。

参考文献:

- [1] MACARIO B A, MICHEL M, MOLINE Y, et al. A comprehensive survey of visual slam algorithms [J]. *Robotics*, 2022, 11 (1): 24.
- [2] ENGEL J, SCHÖPS T, CREMERS D. LSD-SLAM: large-scale direct monocular SLAM [C] //European Conference on Computer Vision, Springer, Cham, 2014: 834 - 849.
- [3] ENGEL J, STÜCKLER J, CREMERS D. Large-scale direct SLAM with stereo cameras [C] //2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE, 2015: 1935 - 1942.
- [4] ENGEL J, KOLTUN V, CREMERS D. Direct sparse odometry [J]. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017, 40 (3): 611 - 625.
- [5] FORSTER C, PIZZOLI M, SCARAMUZZA D. SVO: fast semi-direct monocular visual odometry [C] //2014 IEEE International Conference on Robotics and Automation (ICRA), IEEE, 2014: 15 - 22.
- [6] LI Y, FAN S, SUN Y, et al. Bundle adjustment method using sparse BFGS solution [J]. *Remote Sensing Letters*, 2018, 9 (8): 789 - 798.
- [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. IEEE, 2007: 225 - 234.
- [8] 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.
- [9] GÁLVEZ-LÓPEZ D, TARDOS J D. Bags of binary words for fast place recognition in image sequences [J]. *IEEE Transactions on Robotics*, 2012, 28 (5): 1188 - 1197.
- [10] MUR-ARTAL R, 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.
- [11] CAMPOS C, ELVIRA R, RODRÍGUEZ J J G, et al. ORB-SLAM3: An accurate open-source library for visual, visual - inertial, and multimap slam [J]. *IEEE Transactions on Robotics*, 2021, 37 (6): 1874 - 1890.
- [12] PUMAROLA A, VAKHITOV A, AGUDO A, et al. PL-SLAM: Real-time monocular visual SLAM with points and lines [C] //2017 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2017: 4503 - 4508.
- [13] GOMEZ-OJEDA R, MORENO F A, ZUNIGA-NOËL D, et al. PL-SLAM: a stereo SLAM system through the combination of points and line segments [J]. *IEEE Transactions on Robotics*, 2019, 35 (3): 734 - 746.
- [14] RUBLEE E, RABAUDE V, KONOLIGE K, et al. ORB: an efficient alternative to SIFT or SURF [C] //2011 International Conference on Computer Vision, IEEE, 2011: 2564 - 2571.
- [15] YANG B, WANG Z, YU X. Extraction of ORB features with an improved method [C] //Proceedings of the 3rd International Conference on Vision, Image and Signal Processing, 2019: 1 - 6.
- [16] CALONDER M, LEPETIT V, STRECHA C, et al. Brief: binary robust independent elementary features [C] //European Conference on Computer Vision. Springer, Berlin, Heidelberg, 2010: 778 - 792.
- [17] LUCAS B D, KANADE T. An iterative image registration technique with an application to stereo vision [C] //IJCAI81: 7th International Joint Conference on Artificial Intelligence, 1981, 2: 674 - 679.
- [18] 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.
- [19] LEUTENEGGER S, LYNNEN 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.
- [20] SIBLEY G, MATTHIES L, SUKHATME G. Sliding window filter with application to planetary landing [J]. *Journal of Field Robotics*, 2010, 27 (5): 587 - 608.
- [21] GEIGER A, LENZ P, STILLER C, et al. Vision meets robotics: The kitti dataset [J]. *The International Journal of Robotics Research*, 2013, 32 (11): 1231 - 1237.