

点线特征融合的单目视觉里程计

袁梦, 李艾华, 郑勇, 崔智高, 鲍振强

火箭军工程大学作战保障学院, 陕西 西安 710025

摘要 为了解决地下工程场景下巡逻机器人的定位与建图问题,提出了一种基于点线特征融合的单目视觉里程计(SVO)算法。本文算法可分为特征提取、状态估计和深度滤波器3个线程。特征提取线程负责图像点、线特征的提取;状态估计线程利用点、线特征不同的匹配与跟踪策略获得相机的6自由度位姿,并通过帧与帧、特征与特征、局部帧之间的约束关系进一步优化相机位姿;而深度滤波器线程通过概率分布的方式刻画三维路标点相对于相机光心的深度信息,该方式相对于固定深度值的方式能够提高深度估计的稳健性。本文算法在Euroc公开数据集运行的平均定位精度相对于LSD-SLAM算法提高了17.6%,而在Tum公开数据集上运行的平均定位精度相对于SVO算法提高了6.4%。利用加载摄像头的机器人平台进行测试,实际运行的定位误差大约为1.17%,满足实际需求。

关键词 机器视觉; 巡逻机器人; 自主定位与建图; 点、线特征融合; 视觉里程计; 深度滤波器

中图分类号 TP24

文献标识码 A

doi: 10.3788/LOP55.021501

Point-Line Feature Fusion in Monocular Visual Odometry

Yuan Meng, Li Aihua, Zheng Yong, Cui Zhigao, Bao Zhengqiang

Institute of War Support, Rocket Force University of Engineering, Xi'an, Shaanxi 710025, China

Abstract A semi-direct monocular visual odometry (SVO) algorithm with point-line feature fusion is proposed to solve the problem of localization and mapping in underground engineering for patrol robot. The proposed algorithm is divided into feature extraction, state estimation and depth filter. The point-line feature of image is extracted in the feature extraction thread. The camera pose with 6 degrees of freedom is obtained with different matching and tracking strategies of point-line feature, and it is further optimized by the constraint relationships between frame and frame, feature and feature, and local frames. And the depth information from three-dimensional landmarks to the camera optical center is described through the depth of filter threads with probability distribution. The proposed method can improve the robustness of depth estimation with respect to the fixed depth values. The average positioning accuracy of the proposed algorithm increases by 17.6% on the Euroc dataset compared with that of the LSD-SLAM algorithm, and increases by 6.4% on the Tum dataset compared with that of SVO algorithm. We adopt the robot camera platform to test, and the actual positioning error of about 1.17% meets the actual requirements.

Key words machine vision; patrol robot; autonomous localization and mapping; point-line feature fusion; visual odometry; depth filter

OCIS codes 150.5758; 100.2960; 100.3008; 100.3010

1 引言

单目视觉里程计因其视觉信息丰富且运算量相对较少,而成为基于视觉的及时定位与建图技术

(V-SLAM)的重要分支。实际地下工程场景特征较为稀疏,基于特征的视觉里程计算法(如ORB-SLAM^[1])无法满足实际需求。而基于最小像素强度差的直接法能够有效克服特征比较稀少的室内场

收稿日期: 2017-07-09; **收到修改稿日期:** 2017-08-10

基金项目: 国家自然科学基金青年基金(61501470)

作者简介: 袁梦(1993—),男,硕士研究生,主要从事机器视觉、机器人方面的研究。E-mail: 824035165@qq.com

导师简介: 李艾华(1966—),男,博士,教授,主要从事机器视觉、机器人方面的研究。E-mail: L863@163.com

景以实现精准的定位。Forster 等^[2]提出了一种基于 Fast 特征的半直接单目视觉里程计(SVO)算法,该算法未采用特征描述子匹配,因此计算速度大大提升,但匹配相似度函数的非凸性增大了误匹配的可能性。Engel 等^[3]在 LSD-SLAM 的基础上提出了直接稀疏里程计(DSO)^[4],该算法指出利用光度标定建立精细的相机成像模型可提高定位精度,但是对相机的要求大大提升。

本文从地下工程运用背景出发,综合效率和成本,提出了一种基于点线特征融合的半直接单目视觉里程计算法,有效提高了地下工程巡逻机器人自主定位的准确性和稳健性。

2 算法框架描述

半直接视觉里程计算法可分为 3 个线程:特征提取、位姿估计和深度滤波器。特征提取线程可以

提取输入图像的快速角点和 Canny 边缘特征。位姿估计线程采用光流法进行初始化,并利用直接稀疏法求解相机位姿。根据特征点共视情况,对 2 个关键帧和局部多个关键帧的位姿进行优化,得到更加准确的位姿估计。深度滤波器线程是用纯高斯概率分布来刻画两帧图像之间的三角化,从而得到的深度值和稳健性更强的三维(3D)路标的坐标值,进而加载到地图中供位姿估计使用。由于单目视觉里程计存在着尺度漂移的问题,可采用计算相似变换进行校正。图 1 为本文算法流程图。

视觉里程计包括前端位姿估计与后端位姿优化。位姿估计的核心是特征提取与匹配跟踪,同时需要考虑因单目没有尺度而进行相机初始化、关键帧选择以及特征丢失时进行重定位。位姿优化主要考虑帧与帧、局部帧之间的位姿优化问题,可采用由深度估计收敛的 3D 路标点和关键帧的位姿组成的稀疏地图。

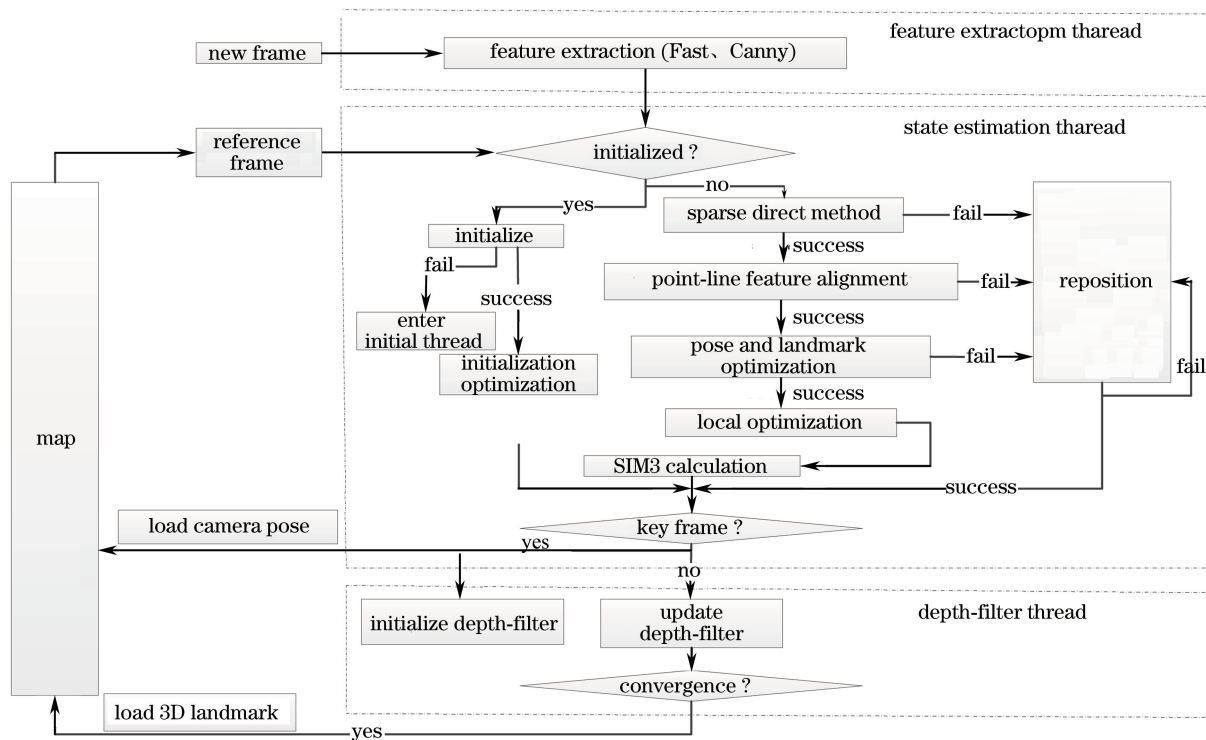


图 1 单目视觉里程计算法流程框图

Fig. 1 Flowchart of monocular visual odometry

3 位姿估计

3.1 特征提取

常用图像特征分为点特征和线特征。点特征能够有效体现图像中特征明显的区域,但是不能很好描述环境信息;而线特征更加侧重图像中的几何约束关系,两者的融合能够更好地描述环境特征信息。

点、线特征提取效果如图 2 所示,其中红色为线特征,绿色为点特征。

由图 2 可知,线特征主要集中在几何特征(梯度)更显著的地方(如轮廓、边角);而点提取的区域更加广泛,其区域分布不具有明显的特征,并且仍存在很多非真正角点,如电脑的轮廓边,这些角点信息会随着相机角度的变化而容易丢失。点线特征融合



图2 TUM数据集中点特征(绿色)和线特征(红色)的提取

Fig. 2 Feature extraction of points (green) and lines (red) in TUM dataset

的方式具有特征局域明显、特征点数目多和提取速度快的优点,有利于提高视觉定位系统的精度。

3.1.1 点特征检测

目前,图像的角点在机器视觉领域得到广泛的运用。图像的角点一般包括图像的极值点、线段的终点和曲线曲率最大的点。角点检测中比较经典的算法有 Harris 角点和 Fast 角点,但是 Harris 角点检测算法需要计算图像像素的导数,从而增加了检测算法的复杂度,降低了整个 SLAM 系统的实时性。相对而言,Fast 角点检测算法的优势在于运算速度快。因此,采用 Fast 角点检测算法作为点特征检测策略。

3.1.2 线特征检测

边缘检测算法分为一阶、二阶算法,其中 Canny 算子是最常用的边缘检测算法。Canny 设计了一种最优预平滑滤波器,使得边缘检测能够更好地定位和检测,从而提供更加精确的线特征信息。为此,采用 Canny 算子作为线特征检测方法。需要注意的是,提取到的线特征是一个带梯度信息的点线,而不是直线段。由图 1 可以看出,特征提取过程会出现特征聚集的情况。因此,采用划分像素区域提取特征的办法使点、线特征分布更加均匀。

3.2 匹配与检测

匹配环节采用直接法中的特征块匹配方法进行特征匹配,其匹配速度快、稳健性较好,并且加入线特征可以有效提高特征不明显区域的匹配准确性。其核心是通过极线搜索找到匹配特征块的粗略区域;采用的特征块相似度函数[本文采用归一化互相关(NCC)函数]最小以寻找最优的匹配位置。相似度函数的非凸性使得特征匹配位置产生误差,而线特征的梯度方向可以有效改善特征匹配精度。

跟踪环节采用经典的稀疏直接法来估计 6 自由度相机位姿。稀疏直接法是利用深度滤波器估计参

考帧上二维(2D)点线特征与对应的 3D 路标的坐标求解出当前帧的位姿。其核心是最小化 3D 路标点 在 2 帧之间投影像元的光度强度误差为

$$\min_{\xi} E(\xi) = \| \mathbf{e} \|_2^2 = \sum_{i=1}^N \mathbf{e}_i^T \mathbf{e}_i, \quad (1)$$

而光度误差 \mathbf{e}_i 为

$$\mathbf{e}_i = \mathbf{I}_c \left(\frac{1}{Z_{ci}} \mathbf{K} \mathbf{P}_i \right) - \mathbf{I}_r \left\{ \frac{1}{Z_{ri}} \mathbf{D} \mathbf{K} [\exp(\hat{\xi}) \mathbf{P}_i] \right\}, \quad (2)$$

式中 \mathbf{I}_c 、 \mathbf{I}_r 分别为当前帧和参考帧的光度值; Z_{ci} 、 Z_{ri} 分别为空间 3D 点 \mathbf{P}_i 在当前帧和参考帧中的深度; \mathbf{K} 为相机内参矩阵; \mathbf{D} 为取前三行的矩阵。需要注意, \mathbf{P}_i 与李代数 $\exp(\hat{\xi})$ 位姿相乘时是齐次坐标的表达。利用 g^2o 库^[5] 中的 Levenberg-Marquardt (L-M) 算法优化求解(1)式,获得相机的位姿。

对于一些大运动位姿估计,需要加入金字塔处理。对图像进行金字塔分级,把上一层的位姿估计结果作为下一层的初始值,这样可以有效地提高大运动造成图像模糊情况下的稳健性。

3.3 相机初始化

采用光流法(LK)追踪初始微小变化的相机位姿。光流法出现误匹配的概率相对较小,而且相对于描述子匹配,其运算速度提升了很多。实际运用中,需要在初始化时相机发生较小的位姿变化,才能应用光流法。

初始化过程中,通过光流法确定参考帧与当前帧上的点、线特征之间的对应关系之后,利用对极几何约束求解当前帧的位姿。经典的做法有:1) 利用本质矩阵分解获得相机位姿;2) 利用单应矩阵分解获得相机位姿。单应矩阵主要应用于特征点集中在同一个平面上(如墙面、地面等),常见于无人机携带俯视相机中。综合考虑场景为近似平面或者非平面、低视差与高视差的情况,提出一种自动切换求解位姿模型的方法。根据光流法跟踪前后像素强度差值的均方差来确定初始化模型,均方差越小,说明跟踪前后的视差较小,可以近似为平面模型,则采用单应矩阵模型,反之则采用本质矩阵模型。

如图 3 所示,当 3D 路标点 p' 并不处于一个平面上时,错误使用单应矩阵模型估计位姿时,会造成特征像素点匹配位置误差(如 u_{ci} 、 u'_{ci}),造成深度估计不准。而当场景比较远的时候,特征像素点匹配位置误差并不大,路标点可以近似在一个平面上。

3.4 关键帧选择

选择合适的关键帧有利于减少处理运算量,如果关键帧选取过少,则估计相机位姿中的 3D 路标

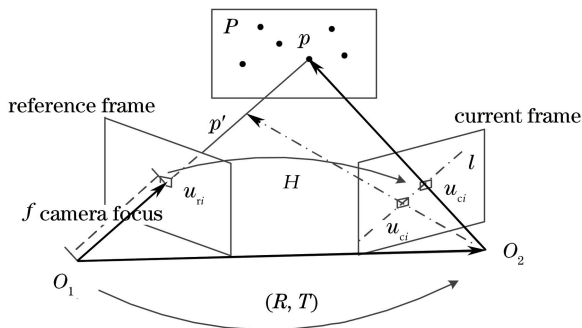


图3 不同初始化模型的误差分析示意图

Fig. 3 Schematic of error analysis for different initialization models

点到当前帧上的投影点会减少,使得求解迭代的精度降低。在综合考虑平视、俯视与旋转等场景,设置基于相机位置变换差异大小和基于相机旋转变换差异大小的2种关键帧选择策略,相对于依靠投影跟踪质量来确定关键帧策略(ORB-SLAM),能够有效提高算法在特征稀少场景下的稳健性。

3.5 重定位

特征跟踪丢失的情况下,重定位环节能够重新确定相机位姿。考虑到单目视觉里程计为轻量级,因此采取将初始化环节的约束条件放宽,并重新初始化,把位姿丢失处的位姿作为初始化的初始位姿。这种方法虽然会使精度有所降低,但是可以有效提高特征丢失后继续定位的能力。

4 位姿优化

由于相机位姿估计图像块的匹配不精确、3D点深度估计不准确以及光照变换等,相机位姿估计不准确。针对上述问题,需要对所估计的相机位姿进行优化。位姿优化分为3个部分,即当前帧与参考帧之间的优化、局部相机位姿优化和全局相机位姿优化。

4.1 两帧之间的优化

4.1.1 初始化位姿优化

初始化过程中,相机畸变参数不精确等问题会使光流法跟踪得到的特征点的位姿存在偏差,其求解出来的位姿也存在误差,为此需要进行初始化位姿优化。采用经典的最小化重投影误差模型优化初始化位姿,同时考虑核函数能有效抑制外点对优化结果的影响,可采用Huber误差模型,其表达式为

$$\xi^* = \underset{\xi}{\operatorname{argmin}} \frac{1}{2} L_{\delta} \left[\sum_{i=1}^n \left\| \mathbf{u}_i - \frac{1}{s_i} \mathbf{K} \exp(\hat{\xi}) \mathbf{P}_i \right\|_2^2 \right], \quad (3)$$

式中 $L_{\delta}(\cdot)$ 为Huber损失函数; \mathbf{u}_i 为当前帧上的点线特征; s_i 为尺度; \mathbf{P}_i 为点线特征对应的3D路标点。采用 g^2_0 中的L-M算法优化(3)式,获得相机的优化位姿。

4.1.2 点线特征校正

考虑到直接法中块匹配存在误匹配、3D路标深度估计不准的情况,需要对投影的特征点位置进行优化,并且参考帧与当前帧之间可能存在旋转,加入仿射变换,其计算方式在PTAM算法^[6]中有详细介绍。当找到所有可被当前帧观测到的3D路标点,将所有3D路标点对应帧上的2D像素块位置经仿射变换后投影到当前帧,通过优化仿射投影到当前帧的像素块位置 \mathbf{u}_i 与当前帧上3D路标点对应的像素块位置 $\hat{\mathbf{u}}_i$ 之间的光度差,求取更加精确的位姿。针对点特征的误差模型的表达式为

$$\hat{\mathbf{u}}_i = \underset{\mathbf{u}_i}{\operatorname{argmin}} \frac{1}{2} \sum_i \left\| \mathbf{I}_c(\hat{\mathbf{u}}_i) - \mathbf{I}_r[A(\mathbf{u}_i)] \right\|_2^2, \quad (4)$$

式中 A 为仿射变换函数。

优化最小二乘问题的方式有前向组合算法和后向组合算法^[7]。后者是一种反向求导方式,计算图像的雅可比式在参考帧图像上进行估计,而不是在当前帧上估计。这些处理方式的优点是只计算一次雅可比式,在一定程度上减少计算量。为此,可以采用后向算法求解该误差模型。

采用 $\mathbf{u}_i \leftarrow \mathbf{u}_i + \Delta \mathbf{u}$ 的更新方式进行更新。为了得到 $\Delta \mathbf{u}$,计算(3)式对 $\Delta \mathbf{u}$ 的偏导数,并将其置为零。

$$\Delta \mathbf{u} = \mathbf{H}^{-1} \cdot \sum_{\mathbf{u}_i} \left(\nabla \mathbf{I}_r \cdot \frac{\partial A}{\partial \mathbf{u}_i} \right) \times \left\{ \mathbf{I}_c(\mathbf{u}_i) - \mathbf{I}_r[A(\mathbf{u}_i)] \right\}, \quad (5)$$

$$\mathbf{H} = \sum_{\mathbf{u}_i} \left[\nabla \mathbf{I}_r \cdot \frac{\partial A}{\partial \mathbf{u}_i} \right]^T \left[\nabla \mathbf{I}_r \cdot \frac{\partial A}{\partial \mathbf{u}_i} \right]. \quad (6)$$

此处需要注意区分点、线特征迭代更新的方向不同。对于点特征,分别从 x 、 y 方向同时进行迭代更新,而线特征的更新方向则是沿着边的方向进行迭代。优化之前, \mathbf{H} 是可以提前计算出来,从而避免了反复计算 \mathbf{H} 。同时,特征校正过程中,点、线特征的位置不再满足对极几何约束,因此需要对地图中的3D路标点和关键帧进行位姿校正。

4.1.3 相机位姿和路标点位置校正

位姿校正是经典的光束法平差过程,通过上一步得到的更加精确的投影点位置来优化地图中的3D路标点位置和关键帧的位姿进行优化。通过最小化投影误差构建误差函数:

$$[T_k, P_i] = \operatorname{argmin}_{T_k, P_i} \frac{1}{2} \sum_i \|u_i - \pi(T_k, P_i)\|^2, \quad (7)$$

式中 π 为空间点到二维图像点的映射函数。

g^2o 库中对于该光束法平差过程有完整的实现。采用 L-M 算法优化误差函数(7)式。

4.2 局部优化与尺度漂移校正

局部优化是视觉里程计中常用于消除累计误差的方式,通过优化具有共视关系(共同观测到 3D 路标点)的关键帧位姿获得局部精确的关键帧位姿,同样, g^2o 库中有完整实现,只需要人工加入一些误差边和节点即可。

单目视觉里程计中,估计相机位姿存在 7 个自由度:相机的旋转和位移各包括 3 个自由度,还有 1 个尺度。由于单目视觉里程计中的尺度无法获得,故将尺度统一为 1。采用多帧图像恢复深度信息,会使尺度发生漂移,因此需要计算一个从当前帧到地图关键帧的相似变换^[1],以消除尺度漂移问题。

5 深度滤波器

深度滤波器的核心是极线搜索和块匹配技术,如图 4 所示。通过估计参考图像帧上点线特征所对应 3D 路标的可能深度范围,利用极线约束关系确定出该特征在当前帧上的对应位置范围,即极线。在极线上比较像素差异并寻找当前帧上对应的点线特征位置。由于特征块相似度的非凸性(如 NCC 函数)使得像素点深度的估计不准确,采用估计深度值的分布方式来描述深度值,而非确定的数值。深度估计是一个状态估计问题,可通过滤波器和基于最小二乘的非线性优化 2 种思路进行求解。非线性优化的效果虽然比滤波器的效果好^[8],但是计算量相对较大。采用滤波器方式估计像素特征点深度,可以提高算法的实时性。

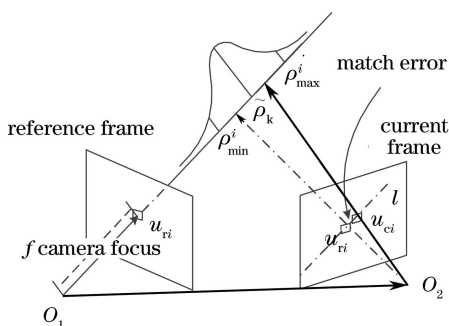


图 4 深度滤波器示意图

Fig. 4 Schematic of depth filter

文献[9]考虑到两帧之间深度估计值的倒数存在最大值 ρ_{\min}^i 和最小值 ρ_{\max}^i 的情况,而这 2 个值并不一定是关于真实值对称的。将深度值的倒数刻画成满足高斯分布+均匀分布的概率分布模型,但是均匀分布并没有改变深度值满足高斯分布的属性,只是增加了对坏点的判断,这样就使深度滤波器模型更加复杂。而采用深度估计值的倒数符合纯高斯分布^[10]的方法来刻画深度滤波器,通过提取的点、线特征匹配光度误差大小检测坏点。

目前,深度的不确定度的刻画方式有:1) 几何不确定度;2) 光度不确定度;3) 两者的融合。为了简便起见,采用几何不确定度,认为极线上存在一个像素大小的误差,如图 4 所示。采用概率分布的方式来描述深度值,当估计的深度值的不确定度小于一定阈值时,则认为该深度值收敛。工程实践中认为,当深度值收敛到一定范围时,则认为所估计的深度值较为准确。

6 实验

6.1 线特征与点特征性能比较

为了验证点、线特征的融合方式能够有效提高特征点匹配跟踪的准确性和稳健性,设计了 2 个性能比较实验,即点、线特征校正和深度估计。

6.1.1 点、线特征校正性能比较

为了比较点、线特征校正性能差异,实验选取同样 TUM RGB-D 数据集^[11]中的 fr2_xyz 进行验证。其实验步骤是:1) 任意选取相机帧作为实验对象,选择时刻“130531102.175304”的数据;2) 认为设定真实值像素位置和误差大小,利用点、线特征校正估计出像素位置,得出与真实位置之间的误差,如表 1 所示。

表 1 点线特征校正性能比较

Table 1 Comparison of correction performance for point-line features

| Experiment 1 | Time consuming /ms | Error / pixel |
|-------------------------|--------------------|---------------|
| Line feature alignment | 1.737 | 0.0059 |
| Point feature alignment | 1.815 | 1.2912 |

由表 1 可以看出,线特征校正的方式相对点特征方式,不仅耗时少,而且精度提高。其主要原因是线特征的梯度信息使极线搜索更加快捷、准确,避免了点特征匹配过程中非线性相似度函数多极点造成误差的现象。因此,特征校正环节可采用点、线特征融合校正的方式,以提高像素匹配过程中的精度。

线特征可以提升匹配精度,点特征可以提高定位的稳健性,不易出现特征丢失的情况。

6.1.2 点、线特征深度估计性能比较

由于点、线特征的性能不同,利用深度不确定度更新迭代和深度的状态也存在差异。选取 TUM RGB-D 数据集^[11]中的 fr2_xyz 数据集来验证点、线深度估计性能。实验步骤为:1) 任意选取相邻 20 帧相机图像,获取第 1 帧姿态和深度值作为参考帧初始化深度滤波器;2) 将剩余相机图片作为当前帧更新深度滤波器中的深度值。估计的深度值与真实值的比较如表 2 所示。

表 2 点线特征深度估计性能比较

Table 2 Comparison of depth estimation performance for point-line features

| Experiment 2 | Time consuming / ms | Convergence number | Average error of depth value / cm |
|----------------------------------------|---------------------|--------------------|-----------------------------------|
| Depth estimation of line feature | 215.33 | 56 | 18.25 |
| Depth estimation of point feature | 277.98 | 286 | 15.89 |
| Depth estimation of point-line feature | 347.97 | 342 | 16.27 |

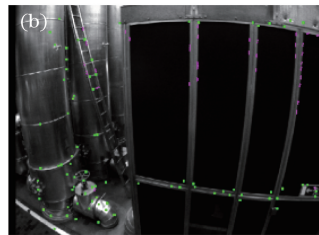
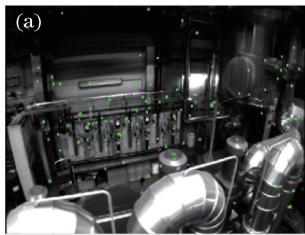


图 5 部分 Euroc 数据集实时运行效果。(a) MH_01 数据集;(b) MH_02 数据集

Fig. 5 Real-time operation effects in part of the Euroc datasets. (a) MH_01 dataset; (b) MH_02 dataset

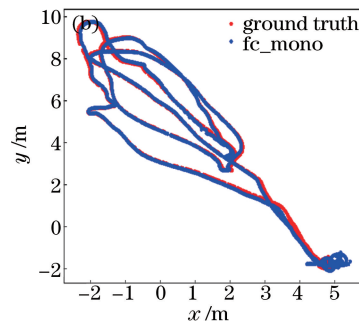
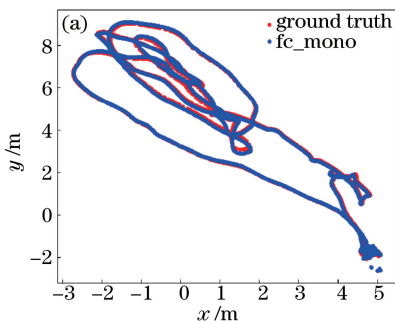


图 6 部分 Euroc 数据集定位效果图。(a) MH_01 数据集;(b) MH_02 数据集

Fig. 6 Location effect in partial Euroc datasets. (a) MH_01 dataset; (b) MH_02 dataset

通过实验 2 可以看出,单纯的线特征深度估计的收敛数目最少,深度值的平均误差最大,因此性能最差。单纯的点特征深度估计的准确性略优于点、线特征融合的方式,说明线特征的融合并不能有效提高深度值的估计准确性,但是增加特征数目有利于提高定位算法的稳健性。

6.2 Euroc 数据集

Euroc 数据集^[12]由无人机挂载的 VI-Sensor 记录的双目图像序列。记录场景包括 2 个不同的房间和 1 个大型的工业环境。图 5 表示 MH_01 和 MH_02 数据集运行过程中的效果图。该数据集的真实路径由 Leica MS50 激光雷达获得,图 5 中红色线段为 Canny 线特征,绿色特征点为 Fast 点特征。图 6 表示 MH_01 和 MH_02 数据集实时运行轨迹和真实轨迹对比效果图。采用绝对位移误差的均方根误差(RMSE)准则评估算法的准确性。

表 3 所示为不同算法的对比结果,其中 SVO、LSD-SLAM 运行结果由文献[9]提供。可以看出,本文算法(fc_mono)的定位精度在 4 个数据集上都要优于 LSD-SLAM,平均提升了 17.6%。前 2 个数据集上的运行效果都要略优于 SVO,但是后 3 个数据集效果则相差较大,主要原因是后 3 个数据集中存在快速的相机移动,致使本文算法在这些快速移动过程中出现了特征丢失的情况。需要说明的是,SVO 只是部分开源(开源版本基本不能在 EUROC 数据集下运行),无法明确后几个数据集定位效果好的原因。

表 3 Euroc 数据集下误差对比实验

Table 3 Error comparison experiment in Euroc dataset

| Experiment 3 | RMSE / m | | |
|--------------|----------|------|-----------------------|
| | fc_mono | SVO | LSD-SLAM (no loop) |
| MH_01 | 0.16 | 0.17 | 0.18 |
| MH_02 | 0.21 | 0.27 | 0.56 |
| MH_03 | 1.70 | 0.43 | 2.69 |
| MH_04 | 2.71 | 1.36 | 2.13 |
| Vicon1_01 | 0.82 | 0.20 | 1.24 |

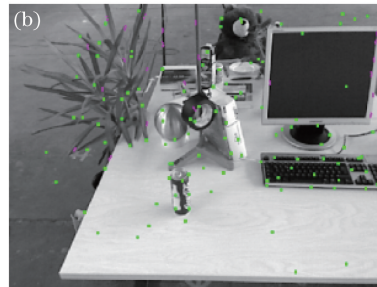
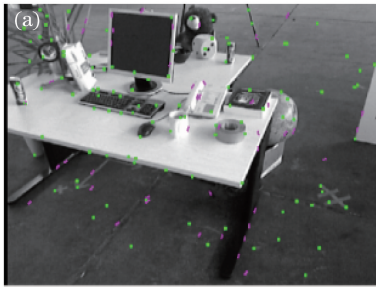


图 7 部分 Tum 数据集实时运行效果。(a) fr2_desk 数据集;(b) fr2_xyz 数据集

Fig. 7 Real-time operation effects in part of the Tum datasets. (a) fr2_desk dataset; (b) fr2_xyz dataset

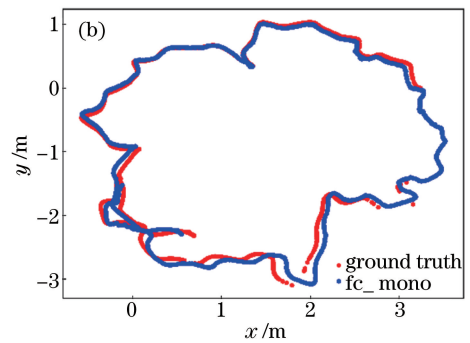
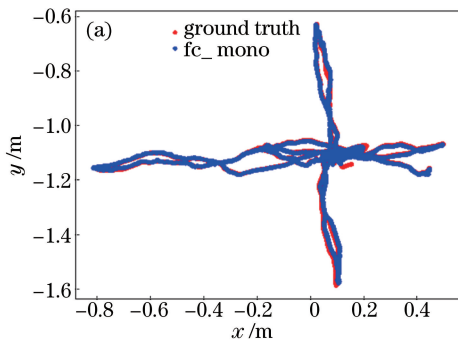


图 8 部分 Tum 数据集定位效果图。(a) fr2_desk 数据集;(b) fr2_xyz 数据集

Fig. 8 Location effects in partial Euroc datasets. (a) fr2_desk dataset; (b) fr2_xyz dataset

从表 4 可以看出,LSD-SLAM 的定位精度较为优异。主要原因在于 LSD-SLAM 采用了闭环检测环节,使得整体定位精度得到提升。本文算法相对 SVO(with edgelets)、semi-dense VO、feature-based RGB-D SLAM 算法,在准确性上分别提高了 6.4%、41.6%、16.5%。

表 4 Tum 数据集下误差对比实验

Table 4 Error comparison experiment in Tum dataset

| Experiment 4 | RMSE / cm | |
|--------------------------|-----------|---------|
| | fr2_desk | fr2_xyz |
| fc_mono | 8.7 | 1.4 |
| SVO (with edgelets) | 9.7 | 1.1 |
| Semi-dense VO | 13.5 | 3.8 |
| Feature-based RGB-D SLAM | 9.5 | 2.6 |
| LSD-SLAM | 4.5 | 1.5 |

6.3 Tum 数据集

Tum 数据集^[11]经常用来评估视觉里程计算方法。该数据集由 Microsoft Kinect RGB-D 相机采集得到,其相片质量相对于 Euroc 数据集较低,原因主要是运动扰动、卷帘快门等。图 7、8 所示分别为 fr2_desk 和 fr2_xyz 数据集运行过程中和实时运行轨迹与真实轨迹对比效果,其中红色线段为 Canny 线特征,绿色特征点为 Fast 点特征。

6.4 实际机器人运行效果

为了验证本文算法的实用性,采用实验室搭建的机器人平台进行验证实验,机器人平台如图 9 所示。摄像头采用免驱 usb2.0 摄像头,帧率为 30 Hz 和卷帘快门。采用摄像头平视进行实验,分为室内场景(特征相对丰富)和走廊场景(特征相对稀少)。

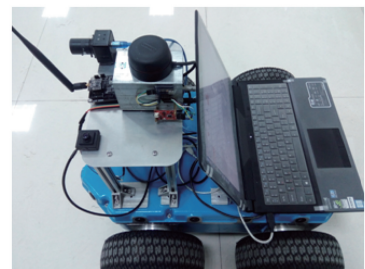


图 9 机器人平台实物图

Fig. 9 Physical map of robot platform

为了验证本文算法的准确性,设计机器人来回一次运行于室内场景中,中间运行路径不一致,最终结束的终点与起点大致相同。室内场景下,估算可得机器人运行的路程约为 34 m。根据图 10(a)比例可以估算出实时估计轨迹起点和终点位置误差约为 0.5 m,而实际终点和起点的误差约为 0.1 m,最终可得到定位误差约为 1.17%,能够满足实际机器人定位需求。

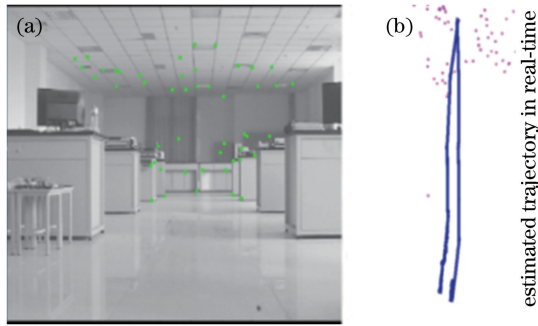


图 10 实际机器人运行效果图。(a)室内场景;(b)走廊场景

Fig. 10 Running effects of actual robot.

(a) Indoor scene; (b) corridor scene

为了验证本文算法的稳健性,设计机器人运行在室内长走廊场景中,机器人实际运行路程约为 55 m,走廊宽度为 2.7 m。由图 10(b)可以看出,机器人运行轨迹基本在一定范围波动,根据比例估算可得到左右波动范围在 0.5 m 以内,符合实际运行的情况。但是,当机器人原地发生较大的旋转,本文算法会进行重定位;当环境特征稀少的时候,本文算法将一直处于重定位状态,无法进行定位。后续将针对原地大旋转情况展开研究,以提高单目视觉里程计定位算法的适用性。

7 结 论

针对特征较为稀少的地下工程场景,提出基于点、线特征融合的半直接单目视觉里程计算法。本文算法在不同公开数据集下得到验证,其准确性比目前部分算法高。在 Euroc 数据集下,本文算法相对于 LSD-SLAM 算法提升了 17.6%。在 TUM 数据集下,本文算法的平均定位精度相对于 SVO (with edgelets)、semi-dense VO、feature-based RGB-D SLAM 算法,在准确性上分别提高了 6.4%、41.6%、16.5%。从实际机器人运行结果可以看出,本文算法不仅能在特征丰富的区域实现准确的定位,误差大约为 1.17%,而且能在特征较为稀少的区域(长走廊)实现较为准确的视觉定位,验证了其准

确性和稳健性。在实际机器人运行过程中,特征丢失的情况较少,不仅因为算法框架的保证,还因为机器人的运动是较为单一和小角度的直线运动。目前,单目视觉定位系统走向实际工程运用仍有一定的距离,实验中也发现大角度的旋转、快速运动等都会出现特征丢失的情况,需要今后进一步的研究。

参 考 文 献

- [1] Mur-Artal R, Montiel J M M, Tardós J D. ORB-SLAM: A versatile and accurate monocular SLAM system[J]. IEEE Transactions on Robotics, 2015, 31(5): 1147-1163.
- [2] Forster C, Pizzoli M, Scaramuzza D. SVO: Fast semi-direct monocular visual odometry [C]. 2014 IEEE International Conference on Robotics and Automation, 2014: 15-22.
- [3] Engel J, Schöps T, Cremers D. LSD-SLAM: Large-scale direct monocular SLAM [C]. European Conference on Computer Vision, 2014, 8690: 834-849.
- [4] Engel J, Koltun V, Cremers D. Direct sparse odometry[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2017, 99: 1-1.
- [5] Kümmerle R, Grisetti G, Strasdat H, et al. g²o: A general framework for graph optimization[C]. 2011 IEEE International Conference on Robotics and Automation, 2011: 3607-3613.
- [6] Klein G, Murray D. Parallel tracking and mapping for small AR workspaces [C]. 2007 6th IEEE and ACM International Symposium on Mixed & Augmented Reality, 2007: 225-234.
- [7] Baker S, Matthews I. Lucas-kanade 20 years on: A unifying framework [J]. International Journal of Computer Vision, 2004, 56(3): 221-255.
- [8] Cadena C, Carlone L, Carrillo H, et al. Simultaneous localization and mapping: Present, future, and the robust-perception age [J]. IEEE Transactions on Robotics, 2016, 32(6): 1-27.
- [9] Forster C, Zhang Z C, Gassner M, et al. SVO: Semidirect visual odometry for monocular and multicamera systems [J]. IEEE Transactions on Robotics, 2016, 33(2): 249-265.
- [10] Civera J, Davison A J, Montiel J M. Inverse depth parametrization for monocular SLAM [J]. IEEE Transactions on Robotics, 2008, 24(5): 932-945.
- [11] 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, 2012: 573-580.
- [12] Burri M, Nikolic J, Gohl P, et al. The EuRoC micro aerial vehicle datasets[J]. The International Journal of Robotics Research, 2016, 35(10): 1157-1163.