

基于点线特征匹配的实时定位及地图重建方法

李运舵^{1,2}, 车进^{1,2*}, 薛澄^{1,2}

¹宁夏大学物理与电子电气工程学院, 宁夏 银川 750021;

²宁夏大学沙漠信息智能感知重点实验室, 宁夏 银川 750021

摘要 目前实时定位及地图重建(SLAM)算法中的实时定位技术已趋于稳定,研究视野转向基于点线特征的半稠密 SLAM 算法。针对此研究方向,提出了一种基于点线特征的 SLAM 算法。首先,在视觉前端使用基于三斑块与局部灰度差分的 oriented fast and rotated brief (ORB) 算法提取特征点并匹配。同时使用 multiscale line segment detector (MLSD) 算法提取线段特征并匹配,使系统在点特征几何变换的基础上加入了线特征的约束条件来计算位姿变换。然后使用点和线约束通过局部 bundle adjustment (BA) 方法优化位姿。最后进行回环检测来重定位。对所提算法在 Euroc 数据集上进行了测试,并与同类算法进行了比较。实验结果表明:所建地图相对较为稠密且轮廓清晰、准确度高;同时在 V1-02-mdeium、V2-02-mdeium、MH-02-easy、MH-03-medium、MH-04-difficult 数据集中均方根误差(RMSE)分别是 0.045、0.0561、0.0539、0.0491、0.0623,相对最低。研究结果表明所提算法在同类算法中具有相对较好的建图效果和较高的精度。

关键词 图像处理; 实时定位及地图重建; 稠密地图; 线特征; 特征匹配; 优化

中图分类号 TP391

文献标志码 A

doi: 10.3788/LOP202259.0210003

Simultaneous Localization and Mapping Based on Point and Line Feature Matching

Li Yunduo^{1,2}, Che Jin^{1,2*}, Xue Cheng^{1,2}

¹School of Physics and Electronic-Electrical Engineering, Ningxia University, Yinchuan, Ningxia 750021, China;

²Ningxia Key Laboratory of Intelligent Sensing for Desert Information, Ningxia University, Yinchuan, Ningxia 750021, China

Abstract At present, the real-time localization technology in simultaneous localization and mapping (SLAM) algorithm has become stable, and the research field turns to the semi-dense SLAM based on point-line feature. Aiming at this research direction, a SLAM algorithm based on point-line feature is proposed. First, an oriented fast and rotated brief (ORB) algorithm based on three patches and local gray difference is used in the visual front end to extract and match feature points. Meanwhile, multiscale line segmentation detector (MLSD) algorithm is used to extract and match line segment features, so that the system adds constraint conditions of line features on the basis of point feature geometric transformation to calculate pose transformation. Then the position and orientation are optimized using point and line constraints through the local bundle adjustment (BA) method. Finally, loop closure detection is performed for repositioning. The proposed algorithm is tested on Euroc dataset and compared with similar algorithms. The experimental results show that the map is relatively dense, with clear outline and high accuracy. At the same time, the root mean square error (RMSE) in V1-02-mdeium, V2-02-mdeium, MH-02-easy,

收稿日期: 2021-02-03; 修回日期: 2021-02-20; 录用日期: 2021-03-09

基金项目: 国家自然科学基金(61861037)

通信作者: *koalache@126.com

MH-03-medium, MH-04-difficult datasets is 0.045, 0.0561, 0.0539, 0.0491, 0.0623 respectively, which is relatively the lowest. The results show that the proposed algorithm has relatively good mapping effect and high accuracy among similar algorithms.

Key words image processing; simultaneous localization and mapping; dense map; line feature; feature matching; optimization

1 引言

实时定位及地图重建(SLAM)一直是无人驾驶领域中的一项基础技术,该技术能够给无人驾驶汽车提供全方位的自动导航、避障等服务。随着时代的进步及生活方式的改变,人们身边逐渐出现了一些集成了SLAM技术的智能机器人,比如导引机器人、清理机器人及服务机器人等。SLAM技术变得大众化,并成为热门的研究方向。许多具有自主导航的机器人也被研发出来,它们在功能上变得更加实用,并且也易于推广。

通过多年的发展和积淀,许多基于SLAM技术的优秀算法被相继提出。文献[1]在2007年提出了第一个实时的单目SLAM(MonoSLAM)算法, MonoSLAM算法的前端部分提取图像的Shi-Tomasi^[2]角点并进行追踪,后端使用拓展卡尔曼滤波(EKF)来完成状态估计,是一种基于稀疏建图的SLAM系统。随着视觉SLAM在理论方面的突破,文献[3]提出了ORB-SLAM算法,该算法前端使用了oriented fast and rotated brief(ORB)的方法对特征点进行提取与匹配,后端采用了非线性优化的方法对位姿进行纠正更新。ORB-SLAM算法的特点是运行速度快、精度高,地图输出的是稀疏点云。文献[4]提出了一种基于ORB特征的改进算法,该算法利用相邻区域特征点的三维空间距离不变性优化视觉里程计。VINS-MONO算法^[5]是一种视觉和惯导融合的算法,它的前端使用了光流追踪特征点,后端对视觉与惯导数据进行融合。且VINS-MONO算法具有自动初始化、重定位、闭环检测及全局位姿图优化功能,系统输出为稀疏点云地图。

上述SLAM算法都是目前非常流行的算法,具有较高的定位精度,可以满足机器人的定位要求。但是对于机器人的导航避障来说,稀疏点云地图无法使机器人进行精确导航避障,更加稠密的地图才能满足需求,因此需要进一步的研究。文献[6]提出了一种基于点线特征的SLAM(PL-SLAM)算法,该算法使用line segment detector(LSD)^[7]提取线特征,利用两帧间线特征具有几何约束的性质,并

结合点特征进行联合位姿计算,最终输出了由点线组成的半稠密地图。文献[8]提出了PL-VINS算法,该算法在VINS-MONO算法的基础上加入了线特征匹配,并与点特征进行联合约束,最终输出半稠密地图。文献[9]提出一种在ORB-SLAM系统框架上,将稠密直接法和稀疏特征法结合在一起求解相机位姿的方法,该方法可生成稠密地图。文献[10]提出一种点线特征结合的位姿估计算法,该算法利用点线约束优化位姿。文献[11]提出了一种改进的半稠密单目ORB-SLAM算法,该算法加入了线特征匹配算法,并提出了关键帧剔除算法以减少冗余帧数目,最终输出半稠密地图。以上的点线SLAM算法使用的线特征算法所提取的线段存在一定噪声,导致连续的线段可能被过度分割;提取的线条比较松散、不连续,降低了线段约束的精度,导致所建地图准确度不高,且地图较为混乱。本文提出了一种使用点线特征进行匹配的SLAM算法,目的是解决点线SLAM所建地图较为混乱的问题,使系统输出精度高、效果好的半稠密SLAM地图,以提升机器人的导航与避障的准确率。

2 基本原理

所提算法的整体框架如图1所示。首先对图像进行点、线特征的提取与匹配,根据它们的匹配关系建立几何约束,并使用随机抽样一致性(RANSAC)算法和line perspective-n-point(LPnP)算法剔除无效的点和线。然后挑选出关键帧,使用点线约束进行局部bundle adjustment(BA)优化位姿,再利用三角化计算出点、线特征的真实坐标,最后进行回环检测重新定位。其中,点特征的提取采用文献[12]中的算法,该算法是基于三斑块与局部灰度差分的改进ORB算法,在特征点的匹配精度上有一定的提升。线特征的提取使用multiscale line segment detector(MLSD)算法^[13],该算法采用多尺度扩展的线段提取器,相对于LSD算法,提取的线段连续性强、质量更好且准确率高。

所提算法是在PL-SLAM算法的基础上改进的。PL-SLAM算法使用ORB与LSD提取点、线特

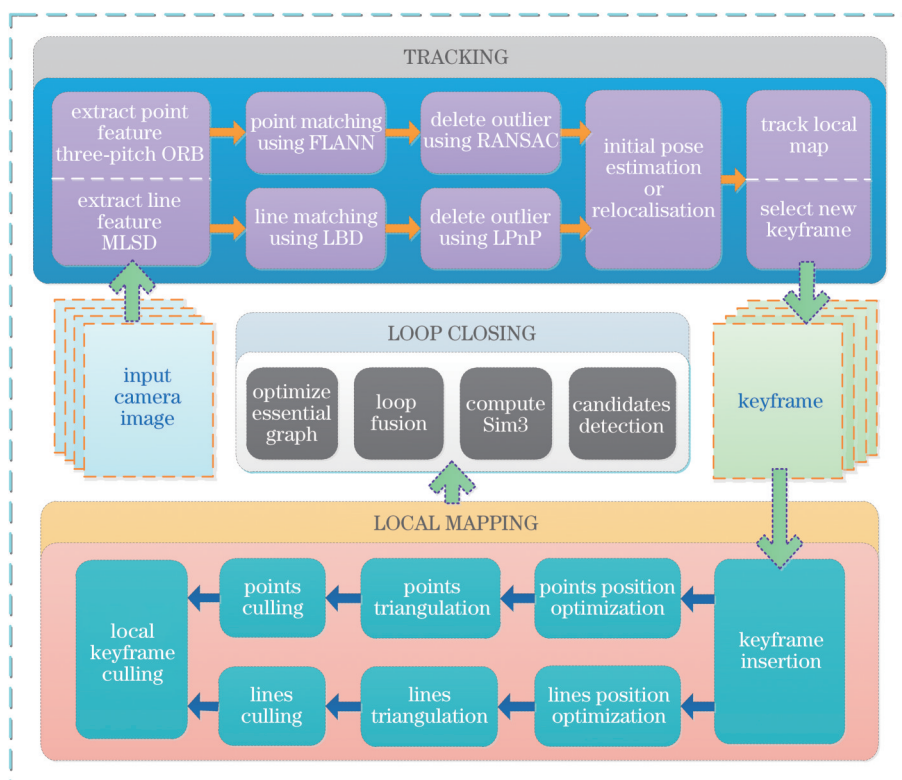


图 1 所提算法框架图

Fig. 1 Frame diagram of proposed algorithm

征,并采用BruteForce Matcher (BFMatcher)方法匹配点特征,利用高斯牛顿法最小化线投影误差。相比之下,所提算法使用了不同的点、线特征提取算法,并在特征匹配上也具有区别。主要改进如下。

1) 使用基于三斑块与局部灰度差分的ORB算法来提取点特征,以减小特征点的误匹配概率,从而提高前端视觉里程计的性能,为后端优化提供可靠数据源。

2) 使用MLS算法来提取线特征,以获得连续性强、质量更好、精度高的线段,建立更加准确的点、线几何约束,提升建图质量。

3) 使用了flann-based matcher (FLANN)算法^[44]进行特征点匹配,提高匹配速度;使用了基于PnP的线段投影法来最小化误差。

3 实现方法

3.1 点特征提取

采用基于三斑块与局部灰度差分的ORB算法来提取特征点,该算法的匹配精度高于ORB算法。与ORB算法一样,基于三斑块与局部灰度差分的ORB算法也提取了features from accelerated segment test (FAST)角点,但由于FAST缺少旋转不变性,

确定特征点的质心方向要结合灰度质心方法。首先定义灰度质心公式为

$$m_{jk} = \sum_{x,y} x^j y^k I(x,y), \quad (1)$$

式中: m 为图像块; $I(x,y)$ 为像素灰度值; x,y 为横纵坐标值,其上角标 j,k 表示幂运算数。图像块 m 的质心坐标 C_m 为

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

特征点与质心的方向夹角 θ_c 为

$$\theta_c = \arctan \left(\frac{m_{01}}{m_{10}} \right) = \arctan \left[\frac{\sum_{x,y} yI(x,y)}{\sum_{x,y} xI(x,y)} \right]. \quad (3)$$

确定了特征点的质心方向后,需要建立特征点描述子来进行特征匹配。原始ORB算法特征点描述子是在候选特征点周围,利用高斯分布随机选取的 n 对像素点(一般 n 取256),该特征点描述子采用了均值滤波来降低噪声敏感度,但会丢失一部分信息,尤其在关键点的高频区域。为了改善这种情况,文献[10]提出了一种三斑块群灰度值比较的方法。

图 2 为三斑块群 ($G_1^{(1)}, G_1^{(2)}, G_1^{(3)}$) 与

$(G_2^{(1)}, G_2^{(2)}, G_2^{(3)})$ 。

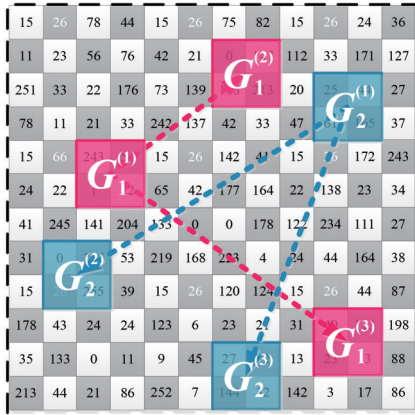


图 2 特征计算示意图

Fig. 2 Schematic diagram of feature calculation

具体步骤为:首先,在特征点周围 $M \times M$ 像素内使用高斯分布随机选择 T 组三斑块群(为了更加直观地展示,图 2 只绘制了两组三斑块群),第 t 组三斑块群 G_t^p 的表达式为

$$G_t^p = (G_t^{(1)}, G_t^{(2)}, G_t^{(3)}) = [(x_t^{(1)}, y_t^{(1)}), (x_t^{(2)}, y_t^{(2)}), (x_t^{(3)}, y_t^{(3)})], \quad (4)$$

式中: $(x_t^{(1)}, y_t^{(1)})$ 、 $(x_t^{(2)}, y_t^{(2)})$ 、 $(x_t^{(3)}, y_t^{(3)})$ 为斑块的坐标,上标为 1 的是主斑块,为 2 和 3 的是副斑块。然

后,分别计算每组主斑块与 2 个副斑块之间的灰度大小关系。只有当 2 个副斑块的灰度均值都小于主斑块时,特征描述符所对应的位为 1,否则为 0。第 t 位描述符的函数 $d_G(\bullet)$ 为

$$d_G(G_t^p) = \begin{cases} 1, & I_G(G_t^{(1)}) > I_G(G_t^{(2)}) \& I_G(G_t^{(1)}) > I_G(G_t^{(3)}) \\ 0, & \text{else} \end{cases}, \quad (5)$$

式中: $I_G(\bullet)$ 为色斑灰度均值函数。将第 1 组到第 T 组的描述符全部计算,最终得到的完整二进制描述符 $D_G(\bullet)$ 为

$$D_G(G_t^p) = \sum_{t=1}^T 2^{t-1} d_G(G_t^p). \quad (6)$$

在特征点匹配时,只需计算描述子的汉明距离,当距离小于阈值时,即可完成匹配。

3.2 点特征匹配

特征点匹配通常采用 BFMatcher 方法,它使用遍历的方式将所有的描述子进行匹配,可以得到最优匹配结果,但是比较耗时。所提系统选择速度更快的 FLANN 算法^[14],它是一种最近邻搜索匹配算法,将所有的描述子通过 K-Dimensional tree (KD-TREE) 的结构进行储存,并且分成几个特定的区域,然后寻找附近邻域内距离最近的描述子,即可就完成特征点的匹配。点特征的提取与匹配如图 3 所示。

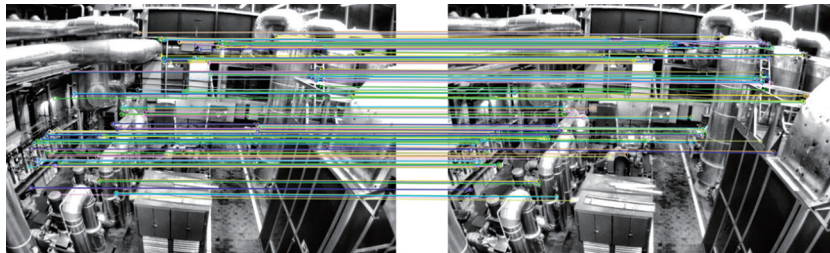


图 3 点特征提取与匹配

Fig. 3 Point feature extraction and matching

3.3 线特征提取

在目前比较流行的点线 SLAM 系统中,线特征提取大多采用 LSD 算法,如 PL-SLAM、PL-VINS 等。也有部分采用 edge drawing lines (EDLines) 算法^[15],如文献^[11],EDLines 算法是一种快速的边缘检测算法,运算速度快于 LSD 算法,精度基本与 LSD 持平。此外还有部分采用 Markov chain marginal line segment detector (MCMLSD) 算法^[16],该算法是一种融合了图像域感知聚类和全局概率霍夫变换的概率算法,可以使用标准动态规划算法精确计算出概率最优解,准确度优于 LSD 算法及

EDLines 算法。但是目前 MCMLSD 算法的源代码仅支持 Matlab 版本,运行效率并不高,因此并不适合运行于实时 SLAM 系统当中。

所提系统采用文献^[13]提出的 MLSD 算法提取线特征。由于具有多尺度特性,MLSD 算法可以使线段不容易出现过分割,对噪声不敏感。并且该算法在不失准确度的同时利用一种逆向框架将一些因噪声或像素间断而被拆散的线段合并在一起,在具有很少特征的场景中匹配线段时,精度仍然比 LSD 算法和 EDlines 算法高。MLSD 算法的详细原理如下。首先计算出图像中每个像素的水平线角

度,可以得到一个单位向量场,使所有向量都与经过基点的水平线相切。然后选择其中一个方向连贯的像素区域,并将该区域定义为一条线段 s 。由于线段 s 由极窄的矩形区域组成,它的宽度可以大于 1 个像素,记 $|s|$ 为线段 s 这个矩形区域的像素面积。基于以上,该算法提出一个假设:若此区域中每个像素的梯度方向都是相互独立的,说明像素之间并无线段相关性,说明无线段存在;反之,如果存在若干个互相关的像素,并且均匀分布在图像中,则说明线段存在。经过计算后,若像素梯度间无相关性,系统会出现一次线段误判警告(FAM)。记 N_{FAM} 为误判警告的数量, N_{FAM} 越大说明相关性越低; N_{FAM} 越小说明相关性越高。设定一个阈值 ϵ ,当 $N_{FAM} \leq \epsilon$ 时,即认为 s 是一条线段。在 $u \times v$ 的图片中(u 和 v 分别代表一个像素块的长和宽), N_{FAM} 为

$$N_{FAM}(S, K, p) = \gamma N_{Lol} \binom{(uv)^{\frac{5}{2}}}{n} \prod_{i=1}^n (|s_i| + 1) \beta(|s_i|, k_i, p), \quad (7)$$

$$\beta(|s_i|, k_i, p) = \sum_{l=k_i}^{|s_i|} \binom{|s_i|}{l} p^l (1-p)^{|s_i|-l}, \quad (8)$$

式中: $\prod_{i=1}^n (|s_i| + 1)$ 表示 S 所包含所有子线段的面积和; N_{Lol} 是图像中线段可能存在的数量; γ 是一个归一化值; S 表示存在一条可能包含 n 条子线段 $\{s_1, \dots, s_n\}$ 的线段; $K = \{k_1, \dots, k_n\}$, k_i 表示在 s_i 中它与 s_i 方向相同的像素数; p 是在矩形区域内的一个随机像素 q 与线段 s_i 的方向 $\theta_s(q)$ 相同的概率; $\beta(|s_i|, k_i, p)$ 是二项分布。

3.4 线特征匹配

提取线特征后需要进行匹配,所提系统采用文献[17]中的线特征匹配方法。该方法利用构成的

line support region (LSR) 线段计算 line band descriptor (LBD)描述符^[18]并进行匹配,然后使用基于PnP的线段投影法来最小化误差^[19]。 PQ 是三维空间中的实际线段, p_d, q_d 是线段端点 PQ 在平面上的投影经过位姿变换矩阵变换后所对应的点,而此时 PQ 两点在平面上的实际投影为 p, q ,如图 4 所示。

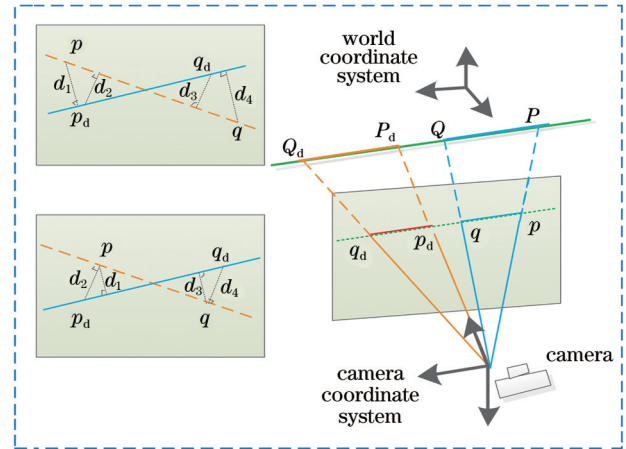


图 4 线投影误差分析图

Fig. 4 Line projection error analysis diagram

线的重投影误差 E_{ir} 为投影线段端点到图像平面中变换后的线段 $p_d q_d$ 之间的距离之和,表达式为

$$E_{ir}(P, Q, I, T, K) = E_{pi}^2(P, I, T, K) + E_{pi}^2(Q, I, T, K), \quad (9)$$

式中:线段 $p_d q_d$ 的归一化线系数 $I = \frac{p_d^z \times q_d^z}{|p_d^z \times q_d^z|}$, p_d^z, q_d^z

是位姿变换后平面上 p_d, q_d 的坐标;通过特征点匹配得到的相机变换矩阵 $T = [R \ t]$; K 表示相机的内参矩阵。最后需要对检测到的重投影线误差进行递归,以便在逼近直线时优化姿态参数 T ,从而完成特征匹配^[20]。线特征的提取与匹配如图 5 所示。

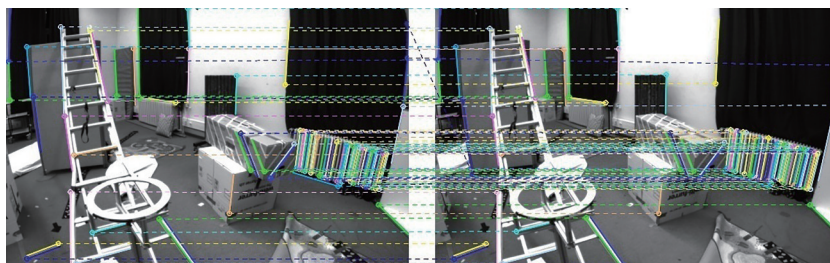


图 5 线特征的提取与匹配

Fig. 5 Line feature extraction and matching

4 实验结果与分析

为了评估所提算法的性能,在文献[21]中提出的标准惯性视觉数据集 Euroc 上进行了相关实验。Euroc 数据集包含多个场景,除了提供双目相机数据及 inertial measurement unit (IMU)数据外,还提供了地面轨迹真值以便验证。因为所提算法采用了点、线特征匹配,所以将所提算法与 PL-SLAM^[6]、PL-VINS^[8]、ORB-SLAM^[3]等算法进行比较,包括点云地图的对比、导航轨迹的对比及系统的各项误差分析。

本测试在台式电脑上进行,配备 AMD Ryzen R7 3700X CPU @3.6 GHz, 16 GB DDR4 内存,采用 Ubuntu 16.04 操作系统,安装了 ROS 机器人操作

系统、OpenCV3、Ceres Solver、G2O 及 Eigen 等软件库,使用 C++ 语言进行编程。

4.1 点云地图对比

为了能够更加直观地展示出所提算法点云地图的效果,在 V1_03_difficult 数据集和 V2_01_easy 数据集上进行了测试。两个数据集的点云平面如图 6(a)、(b)所示,可以作为参照图。图 7 和图 8 分别为 PL-SLAM 算法、PL-VINS 算法、ORB-SLAM 算法、所提算法在以上两种数据集上所建立点云地图。从图中可以看出:PL-SLAM 点云图中的线条比较松散、不连续,地图较为混乱,且点特征较为稀疏,地图轮廓不好分辨;由于使用了光流追踪检测特征点,PL-VINS 可以检测到较为稠密的特征点

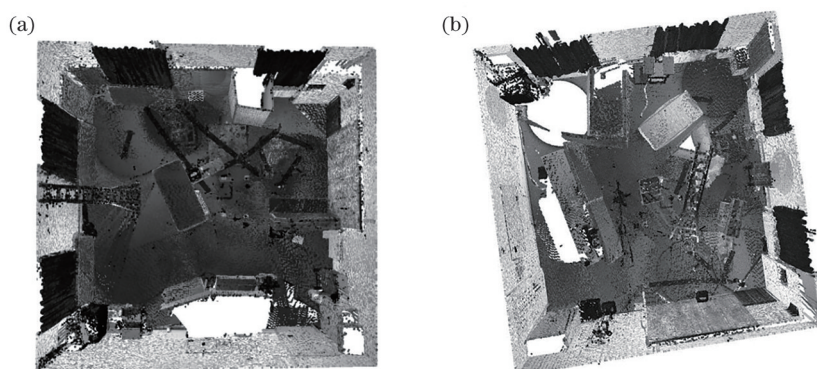


图 6 数据集的点云平面图。(a) V1_03_difficult; (b) V2_01_easy
Fig. 6 Point cloud plan of datasets. (a) V1_03_difficult; (b) V2_01_easy

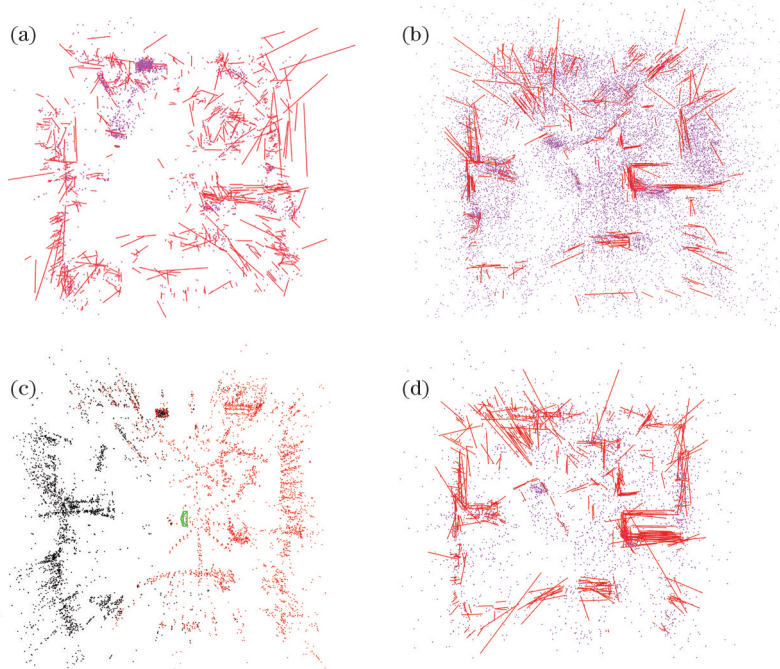


图 7 在 V1_03_difficult 上的测试。(a) PL-SLAM; (b) PL-VINS; (c) ORB-SLAM; (d) 所提算法
Fig. 7 Test on V1_03_difficult. (a) PL-SLAM; (b) PL-VINS; (c) ORB-SLAM; (d) proposed algorithm

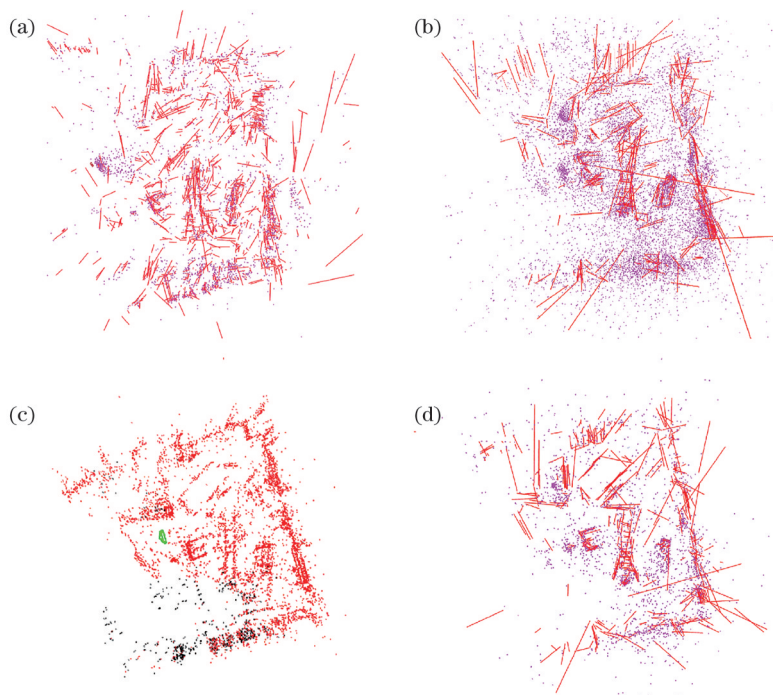


图 8 在 V2_01_easy 上的测试。(a) PL-SLAM; (b) PL-VINS; (c) ORB-SLAM; (d) 所提算法

Fig. 8 Test on V2_01_easy. (a) PL-SLAM; (b) PL-VINS; (c) ORB-SLAM; (d) proposed algorithm

云,但点云图中的线条比较松散、不连续,与参照图相比线条噪声较大,地图轮廓可以勉强分辨;ORB-SLAM的点云图仅由特征点云构成,且点云较为稀疏,轮廓较为模糊;相对于另外几种SLAM算法的点云图,所提算法使用MLSD提取出的线条比较规整、连续性强、准确度高,与参照图比较,地图轮廓及场景内的物体均可以观察到,且辨识度高。综上所述,相对于同类算法,所提算法建立出的点云地图质量较高。

4.2 导航轨迹对比

为了分析所提算法的导航精度,在 V1_02_medium、V2-02-mdeium、MH-02-easy、MH-03-medium 数据集上与真实轨迹和以上几种SLAM的运行轨迹进行了比较。图9为不同算法的运行轨迹及局部放大图。

从图中可以看出:4种算法的输出轨迹从整体上来看差别并不大,仅在细微处有所差别。通过观察局部放大图可以看到,所提算法的轨迹更加接近于真实轨迹,PL-SLAM、ORB-SLAM和PL-VINS算法次之。

4.3 误差分析

通过计算均方根误差(RMSE)来进行误差分析,RMSE值越小说明算法的整体定位精度越高。在多个数据集下进行实验,实验结果如表1所示,最优结

果用粗体表示。从表中可以看出:在10种数据集中,所提算法在5种数据集上的RMSE值比其他算法低;与PL-SLAM算法相比,所提算法在6种数据集上的RMSE值较小;与ORB-SLAM算法相比,所提算法在5种数据集上的RMSE值较小;与PL-VINS算法相比,所提算法在10种数据集上的RMSE值均较小。通过分析RMSE值可知,所提算法的定位精度与PL-SLAM算法、PL-VINS算法对比具有相对优势,与ORB-SLAM算法对比优势并不突出。但ORB-SLAM算法所建地图只包含稀疏点云,不适用于机

表 1 多种SLAM算法的RMSE对比

Table 1 RMSE comparison of various SLAM algorithms

Dataset	Proposed algorithm	PL-SLAM ^[6]	PL-VINS ^[8]	ORB-SLAM ^[3]
V1-01-easy	0.0472	0.0461	0.0628	0.0387
V1-02-mdeium	0.045	0.0477	0.103	0.0664
V1_03_difficult	0.0693	0.0632	0.1518	0.0643
V2-01-easy	0.069	0.0711	0.0806	0.0293
V2-02-mdeium	0.0561	0.0567	0.1713	0.0702
MH-01-easy	0.0611	0.0577	0.1406	0.0312
MH-02-easy	0.0539	0.0550	0.12	0.0681
MH-03-medium	0.0491	0.0561	0.1801	0.0799
MH-04-difficult	0.0623	0.0645	0.2013	0.0655
MH-05-difficult	0.0739	0.067	0.2561	0.0404

机器人避障导航,而所提算法所建地图为包含点线特征的半稠密点云地图,能够给机器人提供有效避障导航服务,具有一定的优势。所提算法误差的降低得益于采用了精度更高的点、线特征提取算法。基于三斑块与局部灰度差分的ORB算法可以生成更加准确的描述子,MLSD算法可以构建具有连续性、高准确率线段,在此基础上进行点、线匹配可以加强系统约束、提高匹配精度、增强系统的整体性能。

5 结 论

提出了一种使用点线特征进行匹配的SLAM算法。经过验证和测试,并与同类算法进行比较发现,所提算法所建地图相对比较稠密、轮廓清晰、准确度高,且运行轨迹更加接近真实轨迹,在V1-02-mdeium、V2-02-mdeium、MH-02-easy、MH-

03-medium、MH-04-difficult数据集中的均方根误差分别是0.045、0.0561、0.0539、0.0491和0.0623,相对误差最低。实验结果表明:所提算法定位精度较高;建图效果相对较好且准确。此结果得益于所提算法采用的点线特征提取方法,将提取的点线做准确匹配并进行联合约束,提升了前端的精度并降低了系统噪声,且线特征使地图更加稠密。接下来的研究方向是继续改进建图部分的算法,使所建地图更加稠密、地图场景更加精确。

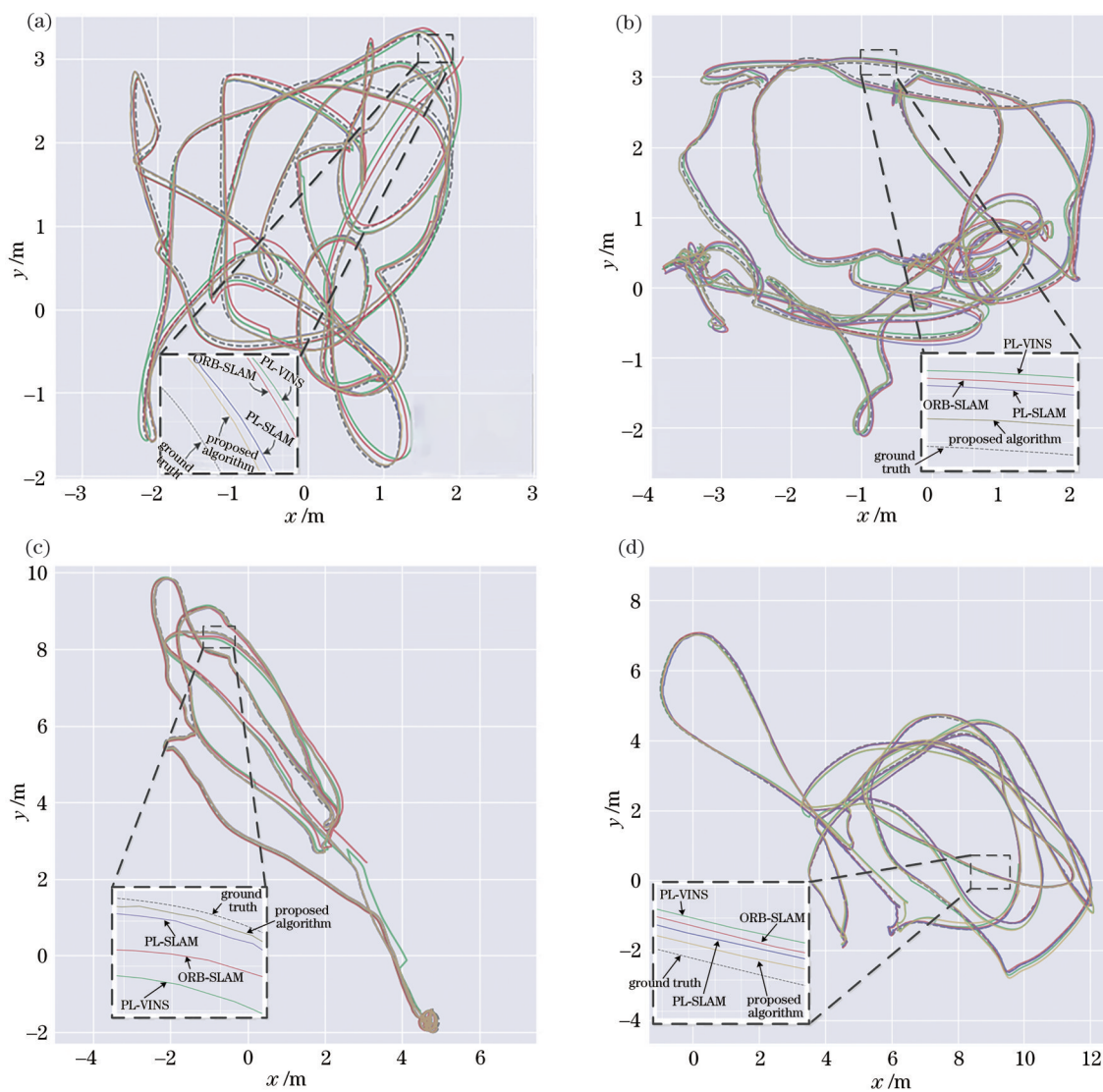


图9 运行轨迹图。(a) V1_02_medium;(b) V2-02-mdeium;(c) MH-02-easy;(d) MH-03-medium
Fig. 9 Running tracks. (a) V1_02_medium; (b) V2-02-mdeium; (c) MH-02-easy; (d) MH-03-medium

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] Shi J B, Tomasi. Good features to track[C]//1994 Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, June 21-23, 1994, Seattle, WA, USA. New York: IEEE Press, 1994: 593-600.
- [3] 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.
- [4] Lin F C, Liu Y H, Zhou J F, et al. Optimization of visual odometry algorithm based on ORB feature[J]. Laser & Optoelectronics Progress, 2019, 56(21): 211507.
林付春, 刘宇红, 周进凡, 等. 基于ORB特征的视觉里程计算法优化[J]. 激光与光电子学进展, 2019, 56(21): 211507.
- [5] 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.
- [6] Gomez-Ojeda R, Moreno F A, Zuñiga-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.
- [7] von Gioi R G, Jakubowicz J, Morel J M, et al. LSD: a line segment detector[J]. Image Processing on Line, 2012, 2: 35-55.
- [8] Fu Q, Wang J L, Yu H S, et al. PL-VINS: real-time monocular visual-inertial SLAM with point and line[EB/OL]. (2020-09-16)[2021-01-25]. <https://arxiv.org/abs/2009.07462>.
- [9] Sun Y L, Wu Q X. An improved simultaneous localization and mapping system[J]. Laser & Optoelectronics Progress, 2019, 56(16): 161013.
孙云雷, 吴清潇. 一种改进的即时定位与地图构建系统[J]. 激光与光电子学进展, 2019, 56(16): 161013.
- [10] Li T, Guan B L, Zhang J M, et al. Relative pose measurement method for monocular cameras by combining the point and line features[J]. Laser & Optoelectronics Progress, 2020, 57(8): 081102.
李涛, 关棒磊, 张家铭, 等. 基于点线特征结合的单目相对位姿测量方法[J]. 激光与光电子学进展, 2020, 57(8): 081102.
- [11] Zhou Y, Kuang H Z, Mou J Z, et al. Improved monocular ORB-SLAM for semi-dense 3D reconstruction[J]. Computer Engineering and Applications, 2021, 57(8): 180-184.
周彦, 旷鸿章, 牟金震, 等. 面向半稠密三维重建的改进单目ORB-SLAM[J]. 计算机工程与应用, 2021, 57(8): 180-184.
- [12] Ma C Q, Hu X G, Xiao J, et al. Improved ORB algorithm using three-patch method and local gray difference[J]. Sensors, 2020, 20(4): 975.
- [13] Salaün Y, Marlet R, Monasse P. Multiscale line segment detector for robust and accurate SfM[C]//2016 23rd International Conference on Pattern Recognition (ICPR), December 4-8, 2016, Cancun, Mexico. New York: IEEE Press, 2016: 2000-2005.
- [14] Hajebi K, Abbasi-Yadkori Y, Shahbazi H, et al. Fast approximate nearest-neighbor search with k-nearest neighbor graph[C]//Proceedings of the 22nd International Joint Conference on Artificial Intelligence, July 16-22, 2011, Barcelona, Catalonia, Spain. Menlo Park: AAAI, 2011: 1312-1317.
- [15] Akinlar C, Topal C. EDLines: a real-time line segment detector with a false detection control[J]. Pattern Recognition Letters, 2011, 32(13): 1633-1642.
- [16] Almazán E J, Tal R, Qian Y M, et al. MCMLSD: a dynamic programming approach to line segment detection[C]//2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), July 21-26, 2017, Honolulu, HI, USA. New York: IEEE Press, 2017: 5854-5862.
- [17] Yao G B, Man X C, Zhang C H, et al. Fully automatic registration of remote sensing images fusion of point and line complementary features[J]. Laser & Optoelectronics Progress, 2020, 57(16): 161503.
姚国标, 满孝成, 张传辉, 等. 融合点线互补特征的遥感影像全自动配准[J]. 激光与光电子学进展, 2020, 57(16): 161503.
- [18] Zhang L L, Koch R. An efficient and robust line segment matching approach based on LBD descriptor and pairwise geometric consistency[J]. Journal of Visual Communication and Image Representation, 2013, 24(7): 794-805.
- [19] Vakhitov A, Funke J, Moreno-Noguer F. Accurate and linear time pose estimation from points and lines[M]//Leibe B, Matas J, Sebe N, et al. Computer vision-ECCV 2016. Lecture notes in computer