

改进点线特征的双目视觉 SLAM 算法

林利蒙, 王梅

(上海大学 机械制造及其自动化系, 上海 200072)

摘要: 针对点线特征 SLAM 算法在图像局部密集区域提取大量相似线特征、同一直线上的线段过度分割等弊端, 提出一种改进点线特征的双目视觉 SLAM 算法 (ISSLAM); 在预处理阶段, 利用梯度密度滤波器剔除图像中特征密集区域, 降低了线特征的误匹配率加速了特征提取过程; 然后, 在 LSD 算法的基础上, 利用线段合并机制, 将同一直线上由于过度分割而产生的断线重新合并, 提高了特征提取的精度; 在闭环检测阶段, 通过融入线特征的扩展词袋模型, 增加了图像相似度计算时的判别依据, 提高了闭环检测的精度; ISSLAM 算法通过增加筛选与合并机制以及扩展的词袋模型, 优化特征提取, 提高特征匹配的效率和精度, 提高算法性能; 最后, 以 EuRoc 公共数据集为实验对象, 证明了算法的有效性。

关键词: 同步定位与地图构建; 点线特征; 梯度密度滤波器; 线段合并; 词袋模型

An Improved Stereo SLAM System through Combination of Points and Line Segments

Lin Limeng, Wang Mei

(Department of Mechanical Manufacturing and Automation, Shanghai University, Shanghai 200072, China)

Abstract: Aiming at the drawbacks of point-line feature SLAM algorithm, such as extracting a large number of similar line features in dense areas of image and over-segmentation of line segments on the same line, a stereo SLAM algorithm (ISSLAM) with improved point-line feature is proposed. In the pre-processing stage, gradient density filter is used to eliminate feature-intensive areas in the image, which reduces the mismatch rate of line features and speeds up the feature extraction process. Then, on the basis of LSD algorithm, line segment merging mechanism is used to merge the broken lines caused by over-segmentation on the same line, improves the accuracy of feature extraction. In the loop closure detection stage, the extended bag-of-words model incorporating line features is used. The criterion of image similarity calculation is added to improve the accuracy of loop closure detection. ISSLAM algorithm improves the efficiency and accuracy of feature matching by adding filtering, merging mechanism and expanding the Bag-of-words model, optimizing feature extraction, and improving the performance of the algorithm. Finally, the EuRoc common data set is taken as the experimental object to prove the effectiveness of the algorithm.

Keywords: SLAM; point and line features; dense-gradient filter; line segment merging; bag-of-words

0 引言

同步定位和地图构建技术^[1] (SLAM) 在过去二十年中一直是计算机视觉和机器人领域的热门研究方向, 被认为是实现移动机器人自主化的核心技术。视觉 SLAM 需要将观测到的图像信息同环境进行数据关联, 其中点特征经常被用于数据关联。

基于点特征的典型算法有 PTAM^[2], SIFT^[3] 和 SURF^[4] 等。由于 SIFT 和 SURF 算法计算量太大, 难以满足实时计算要求, Mur-Artal 等人在 PTAM 算法的基础上提出 ORB-SLAM 算法^[5], 通过快速连续地跟踪 ORB 特征, 以适应实时计算。ORB-SLAM 在纹理丰富的场景中, 有令人印象深刻的表现; 但在处理低纹理场景时, 由于点特征不足或者分布不均而导致算法性能下降。

与点特征相比, 线特征承载更多的结构信息。而且人

造场景中具有丰富的边缘和线性形状的规则结构, 线特征尤为明显。近年来, 基于线特征的算法受到越来越多的关注。Albert Pumarola 等人提出了单目点线结合的视觉 PL-SLAM 算法, 通过端点来表示线段, 并计算端点到线的投影距离来统计误差^[6]。Xiaojia Xie 等人改进了空间线的表示方法, 首次采用正交表示作为最小化参数来模拟线特征以及视觉 SLAM 中的点特征, 并分析得出了关于线参数的重新投影误差的雅可比行列, 改进了 SLAM 的解决方案^[7]。Ruben Gomez-Ojeda 等人又将点线融合的 SLAM 系统扩展到了双目视觉^[8]。

针对部分低纹理场景中点特征不足以及现有 LSD 线特征提取算法存在线段过分割^[9] 问题, 本文提出一种改进点线特征的双目视觉 SLAM 算法 (ISSLAM-Improved Stereo SLAM)。通过梯度密度滤波, 改进 LSD 线特征提取算法; 线段合并机制, 优化提取出的断线; 采用联合点线特征的扩展词袋模型, 提高了 SLAM 算法的精度和鲁棒性。

1 改进点线特征的双目视觉 SLAM 算法

本文所提的改进点线特征的双目视觉 SLAM 算法

收稿日期: 2019-02-12; 修回日期: 2019-03-26。

作者简介: 林利蒙(1993-), 男, 陕西西安人, 硕士研究生, 主要从事机器视觉、同步定位与地图构建方向的研究。

(ISSLAM— Improved Stereo SLAM) 如图 1 所示, 主要包括特征跟踪、局部建图和闭环检测 3 个步骤。首先对输入的图像序列进行预处理, 在此基础上进行点线特征提取, 并通过线段合并来优化点线特征。利用这些点线特征进行特征匹配及相机位姿估计, 并从中筛选出有价值的帧作为关键帧插入到局部地图库中。最后调用场景识别库来进行闭环检测, 优化全局地图。

局部建图调用地图库通过增删等操作进行更新与优化。地图库中的信息主要包括关键帧、地图点和地图直线。场景识别库由点线视觉词典和图像数据库组成, 可以计算图像间的相似程度, 用于判断场景是否形成闭环。

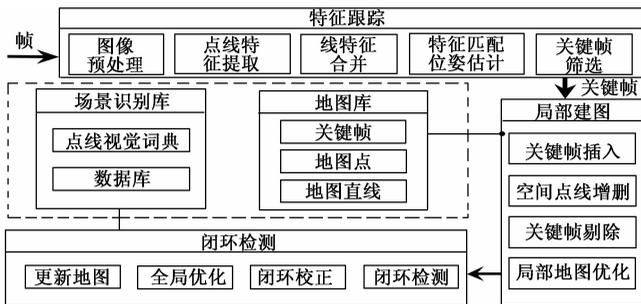


图 1 改进点线特征的双目视觉 SLAM 算法

2 特征跟踪

当下流行的特征跟踪方式通常是基于 ORB 的特征点进行相机运动轨迹的估计, 然而在某些低纹理场景中, 经常由于特征点采集不足而导致跟踪失效。因此我们将线特征加入进来。在线检测方法中, LSD^[9] (line segment detector) 算法被认为是最好和最流行的方法之一。它是按照梯度方向的一致性直接将邻近像素编组为线段区域, 能在线性时间内得出亚像素级精度的检测结果。

虽然 LSD 算法在速度和精度上都取得了不错的效果, 但是也存在自身缺陷。由于它在采集特征时没有筛选与合并机制, 会在图像中的局部密集区域采集大量相似线特征, 且易发生线段过度分割的现象^[9]。

有鉴于此, 本文首先通过梯度密度滤波器^[10]进行图像预处理, 剔除局部密度过高的区域; 然后采用 LSD 算法进行线特征提取, 并且将 LSD 提取出的线段进行合并, 以减少线特征检测的数量, 提高特征检测的精度和效率。

2.1 基于梯度密度滤波器的图像预处理

在我们所检测的环境中, 某些区域中过高的线段检测密度 (如图 2 中所示的网格图案) 通常会导致线匹配的结果不正确且耗费了很多时间。而且在同一密集区域中, 大量相似的线条在匹配过程中也很容易出错。为此, 我们设计了一个滤波器, 利用局部像素梯度的密度值作为筛选准则, 对于特征过于稠密的区域, 以轮廓检测替代完整的线特征检测, 避免出现大量的无效特征, 从而提高特征提取与匹配的效率, 降低算法的计算量与误匹配率。

图像梯度反映了图像中灰度变化的剧烈程度, 线特征



图 2 局部线特征密集区

通常出现在梯度较高的区域。若某区域内梯度较高的像素数量过多时, 意味着该区域高梯度的像素过于密集, 预判为线特征密集区域, 将其剔除以减小后期特征匹配的计算复杂度, 提高匹配精度。

将单位区域内梯度较高的像素点数所占的百分比定义为像素梯度密度, 以衡量该区域是否为特征密集区域, 作为滤波过程中剔除图像中特征密集区域的依据。

设像素点 (i, j) 的梯度值为^[11], 其强度的定义如式 (1) 所示, 当梯度值大于强度阈值时标记为 1, 否则为 0。

$$\tau_{ij} = \begin{cases} 0, & |G_{ij}| \leq G_s \\ 1, & |G_{ij}| > G_s \end{cases} \quad (1)$$

区域的梯度密度 ρ_{ij} 定义为: 以像素点 (i, j) 为中心的 $k \times k$ 区域内, 梯度大于阈值 G_s 的像素点数所占的百分比。

$$\rho_{ij} = \frac{\sum_{i=1}^k \sum_{j=1}^k \tau_{ij}}{K \times K} \quad (2)$$

为了判别梯度密度过高的像素区域, 以进行滤波处理, 减少图像中特征稠密区域对于特征匹配的影响。以像素区域梯度密度阈值 ρ_{DENSE} 为衡量标准, 将梯度密度 ρ_{ij} 大于 ρ_{DENSE} 的像素区域视为特征稠密区域, 线段检测时视为无效区域, 不予处理。由此, 线段检测有效区域标识定义为:

$$D_{ij} = \begin{cases} 1, & \rho_{ij} \leq \rho_{DENSE} \\ 0, & \rho_{ij} > \rho_{DENSE} \end{cases} \quad (3)$$

在滤波过程中, 梯度密度阈值 ρ_{DENSE} 对滤波效果影响很大。阈值设置过大, 无法滤除特征密集区域, 滤波效果不显著; 阈值设置过小, 则会滤除许多有效的检测区域。在此设定 $\rho_{DENSE} = 0.75$ 。

实际滤波过程分检测和标记两个步骤。首先将 k 值设为 5, 找到小范围的高梯度密度区域; 然后在此基础上进行区域扩展, 将 k 值设置为 21, 最终将的区域标记为无效区域, 并对该区域禁用线段检测。



(a) 无梯度密度滤波的检测结果 (b) 梯度密度滤波后的检测结果

图 3 梯度密度滤波前后对比

图 3 (a) 和 3 (b) 分别为 LSD 算法检测的结果和加入

梯度密度滤波器后的效果对比,由此可知,加入滤波器后,密集区域内部不再进行线段检测,而只是在其周围轮廓区域提取部分线段,避免在后期匹配过程中过多细碎分段带来的时间和精度上的影响,减小计算代价,提高系统精度。

利用梯度密度滤波器,剔除了局部的梯度较高的密集区域,最终得到图像中有效的检测区域,为后续点线特征的提取打下基础。

2.2 基于 LSD 线段合并的线特征提取

在线特征提取阶段,利用 LSD 算法提取线特征^[9],在此基础上利用各线段的角度特性、空间位置特性和长度特性,进行断线的分组和合并,解决 LSD 算法过分割的弊端,减少线特征的数量。

LSD 线段合并算法由线段分组和线段合并来完成。在线段分组阶段,利用角度贴近度和空间接近度来判断同一直线上的断线。考虑计算的复杂程度,依次利用角度贴近度、水平坐标接近度和垂直坐标接近度粗略筛选出可合并的同一直线上的候选断线,分为一组。在线段合并阶段,引入线段长度因素,以线段角度、归一化后的长度和空间距离作为可合并性标准,精确判断并完成断线合并。

2.2.1 线段分组

首先,按线段长度的降序对 LSD 算法提取出的线段进行排序,得到 $\{L_1, L_2, L_3, \dots, L_n\}$ 。线段分组从最长的线段 L_1 开始进行,因为较长的线段往往来自具有连续强梯度的图像区域,所以更可靠。

线段 L_i 的两个端点为 (A_i, B_i) , 其坐标分别为 (x_{A_i}, y_{A_i}) 和 (x_{B_i}, y_{B_i}) 。 L_i 的长度为 l_i , 与水平方向的夹角为 θ_i 。

不失一般性,设当前最长线段为 L_1 , 则线段集合 $L = \{L_2, L_3, \dots, L_n\}$ 中满足角度筛选条件式 (4)、水平距离筛选条件式 (5)、垂直距离筛选条件式 (6) 的线段分为一组,构成线段组。

线段分组流程如图 4 所示。

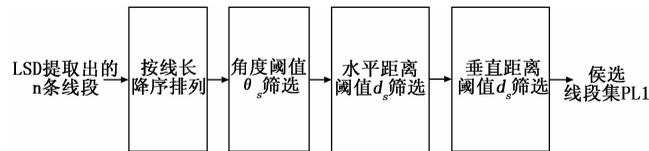


图 4 线段分组流程

经过角度筛选,得到候选线段组 L_1 :

$$L_1 = [L_i \in L : (|\theta_i - \theta_1| < \theta_s)] \quad (4)$$

其中: θ_s 为衡量线段特征角度相近程度的角度筛选阈值。

经过水平距离筛选,得到候选线段组 L_2 :

$$L_2 = [\forall L_i \in L_1 : \left\{ \min \left(\begin{array}{l} |x_{A_i} - x_{A_1}|, |x_{A_i} - x_{B_1}| \\ |x_{B_i} - x_{A_1}|, |x_{B_i} - x_{B_1}| \end{array} \right) < d_h \right\}] \quad (5)$$

经过垂直距离筛选,得到线段组 P_{L1} :

$$P_{L1} = [\forall L_i \in L_2 : \left\{ \min \left(\begin{array}{l} |y_{A_i} - y_{A_1}|, |y_{A_i} - y_{B_1}| \\ |y_{B_i} - y_{A_1}|, |y_{B_i} - y_{B_1}| \end{array} \right) < d_v \right\}] \quad (6)$$

其中: d_s 为衡量线段特征水平和垂直距离接近程度的距离筛选阈值。

依据 3 个筛选条件,将角度和空间上都与 L_1 很接近的线段分为一组,最终得到候选线段集 $P_{L1} = \{L_2, L_3, \dots, L_n\}$ 为后续线段合并打下基础。值得注意的是 3 个筛选条件的顺序很重要,因为角度筛选的计算最简便,所以应该最先使用。此外,以水平距离筛选和垂直距离筛选取代欧几里德距离筛选,由于仅涉及绝对值的加减运算,极大地提高筛选效率,减小计算代价。

2.2.2 线段合并

经过线段分组得到的候选线段集 P_{L1} , 还需要经过端点距离、角度修正值和夹角偏差的进一步精确筛选,以完成线段合并,流程如图 5 所示。

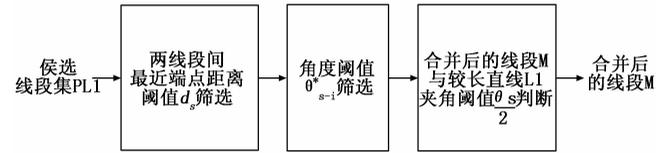


图 5 线段合并流程图

如图 6 所示,两条线段 L_i (端点 A_i, B_i) 和 L_j (端点 A_j, B_j) 的间距,用其最近端点的距离 d_{ij} 表示:

$$d_{ij} = \min(|A_i - A_j|, |A_i - B_j|, |B_i - A_j|, |B_i - B_j|) \quad (7)$$

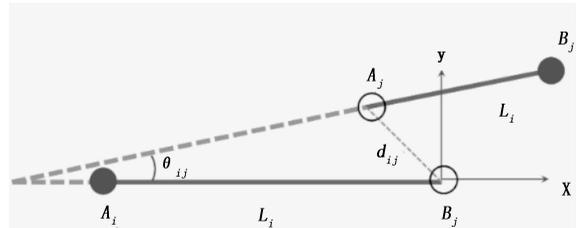


图 6 两线段最近端点距离表示

利用最近端点距离,筛选出两条线段空间接近程度小于阈值 d_s 的候选线段组 L_3 :

$$L_3 = [\forall L_i \in P_{L1} : (d_{ij} < d_s)] \quad (8)$$

考虑到线段长度和间距的影响因素,需对候选线段组 L_3 中的角度阈值做出修正。当前较长线段为 L_1 , 与其相匹配的线段 $l_i \in \{L_2, L_3, \dots, L_s\}$ 的阈值修正系数为 λ_i 为:

$$\lambda_i = \frac{l_i}{l_1} + \frac{1}{\xi_s} \frac{d_{1i}}{l_1} \quad (9)$$

其中: $\frac{1}{\xi_s}$ 为两条线段间距的加权系数, $0 < \xi_s < 1$, 且 $\xi_s = \frac{d_s}{l_1}$ 用以调整线段长度和间距的比例关系。

综合考虑线段长度和两条线段间距对角度筛选的影响,

线段 L_i 的自适应角度阈值为:

$$\theta_{s-i}^* = \left(1 - \frac{1}{1 + e^{-2(\lambda-1.5)}}\right)\theta_s \quad (10)$$

由公式 10 可以看出, θ_{s-i}^* 与线段长度 l_i 成正比, 与其和 L_1 的间距 d_{1i} 也成正比。 l_i 和 d_{1i} 越小, 阈值 θ_{s-i}^* 越大, 可合并性增加。

候选线段组 L_3 中, 若两线段间的夹角 $|\theta_i - \theta_j| < \theta_{s-i}^*$, 则初步满足合并标准, 将其预合并为线段 M 。若预合并后的线段 M 与较长线段 L_1 之间的绝对角度差小于 $\frac{\theta_s}{2}$, 合并成立, 以合并线段 M 替换 L_1 和 L_i ; 否则, 由于合并前后角度差过大, 合并结果偏离原线段, 则放弃合并, 以确保合并的段不会偏离较长的段 L_1 太远。因而, 满足式 (11) 条件的线段进行合并。

$$L_4 = \left[\forall L_i \in L_3: |\theta_i - \theta_1| < \theta_{s-i}^* \text{ 且 } |\theta_M - \theta_1| < \frac{\theta_s}{2} \right] \quad (11)$$

其中: θ_M 是预合并后的线段 M 的倾角。

线段分组及合并这两个步骤, 重复迭代地进行, 直到无法再合并为止。

图 7 (a) 为 LSD 算法检测出的线段, 白色框中表明线段存在断线的情况; 图 7 (b) 为改进点线特征的双目视觉 SLAM 算法 (ISSLAM) 的检测结果。可以看出, ISSLAM 算法将 LSD 检测出的断线很好的合并在一起 (如白线框中区域所示), 提升了系统的精度。

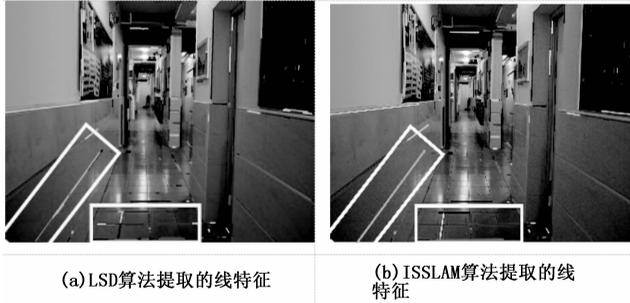


图 7 线段合并效果对比

根据滤波处理、线段检测和合并后得到的线段特征, 计算通用的 LBD^[12] 描述子, 用于线段间的特征匹配和运动估计。

2.3 运动估计

利用图像中提取出的特征点、线的描述子进行特征匹配, 可以得到 3D 特征与 2D 特征的匹配关系。将当前帧观测到的 3D 特征点、线投影到下一帧的 2D 图像平面上, 然后根据相邻帧间投影点线与匹配的线之间的误差关系, 可以进行相机姿态的初步估计^[13]。具体方法不再赘述。

2.4 关键帧选择

相机运动过程中会采集数据, 连续运动一段时间后这些数据便组成了场景的地图。但由于内存资源有限, 我们不可能将所有帧都插入地图中, 而是会从中选择出一些有

代表性的关键帧。

本文沿用 ORB-SLAM 中关键帧选择的策略^[5], 对于当前帧, 首先找到与之具有最多共同观测的关键帧作为参考关键帧, 满足以下所有条件即可视为关键帧:

- 1) 距离上一次关键帧的插入已经过了 20 帧或者局部地图构建线程空闲
- 2) 当前帧至少跟踪到了 50 个特征点和 15 条空间直线
- 3) 当前帧包含参考关键帧中的特征要少于 75%

条件 2 保证了位姿估计的质量, 条件 3 保证了两帧图像之间具有一定的视觉变化。

3 局部建图

当筛选出的新关键帧插入地图后, 要建立当前关键帧与其他关键帧之间的连接关系。在跟踪线程中已经获得了相机位姿和环境地图。在局部建图线程中, 要对位姿和路标进一步优化。局部建图线程主要包括两部分: 局部地图管理及局部地图优化。

局部地图管理的主要任务包括: (1) 把当前处理的关键帧中提取的 2D 点、线特征 3D 参数化到地图数据库中; (2) 对地图中重复的点、线特征进行融合; (3) 对地图中跟踪质量不佳的点、线特征进行剔除; (4) 对地图中的冗余关键帧进行剔除。

局部地图优化就是指从环境中抽取一部分位姿和路标进行优化。因为每一帧都是通过帧之间的相对运动得到的, 所以不可避免的会有误差积累, 故需要优化, 减少误差。

其具体计算方式如下, 首先定义变量, 用 ψ 表示总的投影误差, 由点的投影误差 e_{ij} 和线的投影误差 e_{ik} 组成:

$$\psi^* = \operatorname{argmin}_{\psi} \left[\sum_{i \in K_i} \left(\sum_{j \in P_i} (e_{ij}^T \sum_{c_a}^{-1} e_{ij}) \right) + \sum_{k \in L_i} \left(e_{ik}^T \sum_{c_a}^{-1} e_{ik} \right) \right] \quad (12)$$

其中: K_i 、 P_i 、 L_i 分别代表局部地图中的关键帧数、点特征数、线段特征数。 $\sum_{c_a}^{-1}$ 和 $\sum_{c_a}^{-1}$ 是点、线观测协方差矩阵的逆, 表示投影误差的不确定度。

投影误差 e_{ij} 为 3D 点 X_{oj} 投影到第 i 帧图像上的 2D 点 $\pi(\xi_{io}, X_{oj})$ 与实际像素位置 x_{ij} 之差:

$$e_{ij} = x_{ij} - \pi(\xi_{io}, X_{oj}) \quad (13)$$

其中: π 为投影函数, ξ_{io} 为关键帧 i 的位姿。

线的投影误差 e_{ik} 由线段 k 的端点的 3D 位置 (P_{ok}, Q_{ok}) 在第 i 帧图像上的投影与实际线段所在直线 l_{ik} 计算而得:

$$e_{ik} = \begin{bmatrix} l_{ik} \cdot \pi(\xi_{io}, P_{ok}) \\ l_{ik} \cdot \pi(\xi_{io}, Q_{ok}) \end{bmatrix} \quad (14)$$

采用 Levenberg-Marquardt 方法^[14] 优化投影误差 ψ , 误差的增量为:

$$\Delta\psi = [H + \lambda \operatorname{diag}(H)]^{-1} J^T W e \quad (15)$$

其中的 J 、 H 代表了雅可比矩阵和海瑟矩阵, e 是包含了点和线的总误差, λ 是变量因子, λ 较小时, 接近于高斯牛顿法, 当 λ 较大时, 接近于最速下降法。

$$\psi(k+1) = \psi(k) + \Delta\psi \quad (16)$$

通过迭代计算式 (16) 直到收敛, 求得最小化投影误差 ψ 。

有别于运动估计阶段仅考虑相邻帧间的投影误差, 在局部地图优化阶段, 需要综合考虑局部地图内一系列关键帧间的投影误差和, 通过求解多帧间误差和的最小值, 以调整局部地图中的相机位姿和路标点位置, 从而得到更精确的结果。

4 闭环检测

视觉 SLAM 系统在缺少外部控制信息时, 系统的累积误差会不断增大, 长期积累后会造定位失效。在实际应用中, 相机会运动至先前经过的地方从而形成闭环。闭环的引入可有效的修正累积误差, 对长时间远距离运动的视觉 SLAM 有重要意义。闭环检测实质是判断当前图像与历史较早图像之间是否存在关联, 若两图像相似度达到一定程度, 则认为形成了闭环。基于 BoW 的词袋模型^[15]方法是应用最广泛的闭环检测方法之一, 它从图像中提取特征点, 矢量化成视觉单词, 然后建立词汇树用于图像间相似度的判别。

本文对原有的基于特征点的词袋模型方法进行扩展, 将线特征也考虑进来, 提出了一种新的词汇树构造方式, 提升了回环检测的精度和鲁棒性。

4.1 扩展词汇树的构造方法

传统的词汇树构建方法是基于点特征进行创建的, 需使用大量的点特征描述子进行训练, 以构建视觉词典, 然后不断用 k 均值聚类算法^[16]建立词汇树。

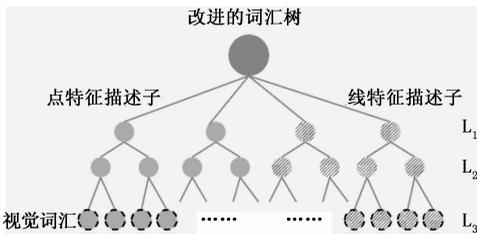


图 8 融合点线特征的词袋模型

尽管已经证明这种方法具有可扩展性和高效性, 但它并不适用于同时具有点线特征类型的词汇树。因此, 本文提出图 8 所示的改进的词汇树。首先将训练的 ORB 点特征描述子和 LBD 线特征描述子放在同一个视觉词典中, 然后将训练的描述符划分为点、线 2 个组, 再分别对点、线特征描述符进行分层 k 均值聚类过程, 最终构造扩展的树结构。

为区分词汇树中不同的特征类型, 在每个特征描述符的末尾附加一个额外字节 (点描述子添加标志位 0, 线描述子添加标志位 1), 以区分其所属的特征类型。在此示例中, 实心填充的是点特征词汇, 线条填充的是线特征词汇, 虚线包围的叶子节点为视觉词典中的视觉词汇。

4.2 融合点线的词包向量

改进后的词包 VPL_i :

$$VPL_i = \{\alpha_1(VP_1, VP_2, \dots, VP_m), \alpha_2(VL_1, VL_2, \dots, VL_m)\} \quad (17)$$

其中, 既包含了点特征词包 $VP_i = (VP_1, VP_2, \dots, VP_m)$, 又包含了线特征词包 $VL_i = (VL_1, VL_2, \dots, VL_m)$ 。 α_1 、 α_2 分别代表点特征和线特征词包所占的权重。针对不同场景的实际特点, 合理设置权值, 调整二者比重。例如纹理丰富的场景中点特征权重值大一些, 在低纹理的环境中让线特征的权重大一些。通常, 权值默认为 0.5。

4.3 词包向量的相似度计算

由于不同词汇的重要程度不同, 需对每个单词赋予不同权重值, 在本文中采用 TF-IDF^[17]进行加权判断。

VPL_i 和 VPL_j 分别代表第 i 帧和第 j 词包向量, 其相似度得分计算如下:

$$s(VPL_i, VPL_j) = 1 - \frac{1}{2} \left| \frac{VPL_i}{|VPL_i|} - \frac{VPL_j}{|VPL_j|} \right| \quad (18)$$

在此, 采用 L_1 范数进行距离计算。

在进行图像相似度判断时, 可直接比较图像词包向量的相似度得分, 降低了计算复杂度提升了检测效率。同时为了避免将相似度较高的邻近帧误判为闭环, 设置了阈值 k , 只有当时 $j > i + k$, 才进行闭环检测。

5 实验与分析

实验数据集来源于 EuRoc 微型飞行器 (MAV) 采集的两个室内场景, 其中包含来自 20FPS 的全局快门相机的立体图像。每个数据集提供由 VICON 运动捕捉系统给出的地面实况轨迹。所有外在和内在参数也由数据平台获取^[18]。

测试平台为华硕 K550D@2.10 Ghz, 12 GB 内存, 64 位 Linux 操作系统。

图 9 (a) ~ (d) 展示由 MAV 采集的室内实景以及提取出的特征点线。

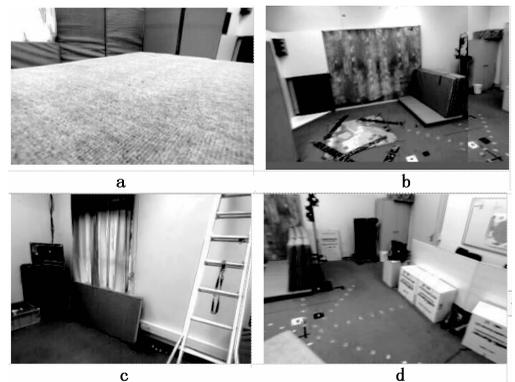


图 9 V1_02_medium 数据集实景

图 10 是由 V1_02_medium 图像序列重建出的地图。图 10 (a) 为基于点特征的 ORB-SLAM2 算法创建的地图, 图 10 (b) 为联合点线特征的文本算法创建的地图。

由 10 (a) 和 10 (b) 的对比可见, 单纯基于点特征所创建出来的地图缺少结构信息, 因此比较杂乱; 而将线特征引入进来后, 所创建的地图结构化信息更多, 从图中的

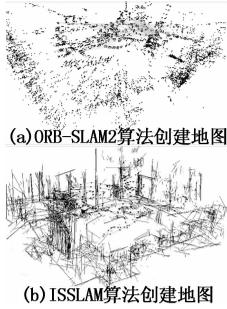


图 10 由 V1_02_medium 重建出的地图

空间直线可以辨别出墙面, 地面和墙的边角等信息。更好地还原了场景结构。表明 ISSLAM 算法在结构化边缘线特征多的人造场景中适用性更高。

由 V1_02_medium 图像序列计算得到的相机运行轨迹分别显示在图 11 (a) 和 11 (b) 中。

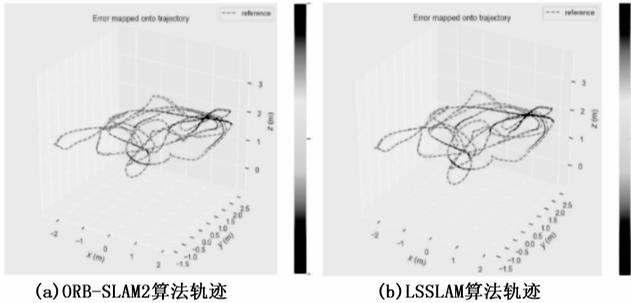


图 11 ORB-SLAM2 算法和 ISSLAM 算法轨迹图对比

图中, 轨迹点上的颜色信息反映了误差大小, 通过颜色与误差数值的对应关系 (图形右侧误差色带) 可以看出, ORB-SLAM2 算法 (图 11 (a)) 的误差区间为 (0.017~0.106), 而本文所提出算法 (图 11 (b)) 的误差区间仅为 (0.003~0.081)。由此可知, 本文所提出的算法运行结果的误差更小, 准确率更高。

为了说明 ISSLAM 算法在不同场景下的有效性与适应性, 分别以 EuRoc 中的多组图像序列 (V1_01_easy、V1_02_medium、V2_03_difficult、MH_02_easy、MH_03_medium、MH_04_difficult) 为实验对象。这些序列分成, 简单、中等、和困难, 这取决于 MAV (微型飞行器) 的速度, 照明和场景的纹理。表 1 为 ORB-SLAM2 算法、点线联合的 PL-SLAM 算法、ISSLAM 算法得到的 RMSE 值^[19] (均方根误差)。

表 1 基于 EuRoc MAV 数据集的运动估计误差

数据集	ORB-SLAM2 算法	PL-SLAM 算法	ISSLAM 算法
V1_01_easy	0.0408	0.0421	0.0395
V1_02_medium	0.0646	0.0462	0.0423
V2_03_difficult	X	0.1263	0.1072
MH_02_easy	0.0636	0.0523	0.0475
MH_03_medium	0.0715	0.0402	0.0365
MH_04_difficult	0.0543	0.0638	0.0593

从表 1 中可以看出, 对于结构化线条较为丰富的室内场景, 联合点线特征的 SLAM 算法鲁棒性更好, 即使在快速运动场景 (V2_03_difficult 测试集), 不会出现 ORB 算法所面临的特征跟踪失败的情况 (表中 X 所示)。而且所提的 ISSLAM 算法比传统的联合点线特征的 PL-SLAM 算法测试的误差更小, 精确度更高。

综上所述, 本文提出的 ISSLAM 算法在精度和鲁棒性上都表现出良好的优势。

6 结语

本文提出一种改进点线特征的双目视觉 SLAM 算法 (ISSLAM) 来进行位姿估计和地图创建。在特征提取过程中, 利用梯度密度滤波器进行图像预处理, 以剔除局部梯度密度过高的区域; 然后采用 LSD 算法进行线特征提取, 并且将 LSD 提取出的线段进行合并, 以减少线特征检测的数量, 提高检测精度。在闭环检测线程中, 对原有基于点特征的词袋模型进行了扩展, 将线特征也引入进来, 提高了算法的精度及鲁棒性。最后, 以 EuRoc 公共数据集为实验对象, 通过与基于点特征的 ORB-SLAM2 算法和基于点线特征的 PL-SLAM 算法比较, 表明了 ISSLAM 算法的有效性。

线特征的引入, 虽然提高了算法精度, 但增加了特征提取与匹配阶段的计算代价, 耗时更长, 不利于算法的实时性, 需要做进一步改进以提高算法效率。

参考文献:

- [1] Durrant-Whyte H, Bailey T. Simultaneous localization and mapping: part I [J]. IEEE Robotics & Automation Magazine, 2006, 13 (2): 99-110.
- [2] Klein G, Murray D. Parallel Tracking and Mapping for Small AR Workspaces [A]. IEEE & Acm International Symposium on Mixed & Augmented Reality [C]. IEEE, 2008.
- [3] Lowe D G. Distinctive Image Features from Scale-Invariant Keypoints [J]. International Journal of Computer Vision, 2004, 60 (2): 91-110.
- [4] Bay H, Ess A, Tuytelaars T, et al. Speeded-Up Robust Features (SURF) [J]. Computer Vision and Image Understanding, 2008, 110 (3): 346-359.
- [5] 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.
- [6] Pumarola A, Vakhitov A, Agudo A, et al. PL-SLAM: Real-time monocular visual SLAM with points and lines [A]. 2017 IEEE International Conference on Robotics and Automation (ICRA) [C]. IEEE, 2017.
- [7] Zuo Xingxing, Xie Xiaojia, Liu Yong, et al. Robust Visual SLAM with Point and Line Features [J]. 2017.
- [8] Gomez-Ojeda R, Zuñiga-Noël, David, Moreno F A, et al. PL-SLAM: a Stereo SLAM System through the Combination of Points and Line Segments [J]. 2017.

[9] Gioi R G V , Jérémie Jakubowicz , Morel J M , et al. LSD: A Fast Line Segment Detector with a False Detection Control [J]. IEEE Transactions on Software Engineering, 2010, 32 (4): 722 - 732.

[10] Johann Salaün , Marlet R , Monasse P. The Multiscale Line Segment Detector [J]. 2016.

[11] Xueshun P , Suqi Z , Junhua G , et al. Improved L0 Gradient Minimization with L1 Fidelity for Image Smoothing [J]. PLOS ONE, 2015, 10 (9): e0138682.

[12] Zhang L , Koch R. An efficient and robust line segment matching approach based on LBD descriptor and pairwise geometric consistency [J]. Journal of Visual Communication & Image Representation, 2013, 24 (7): 794 - 805.

[13] Vakhitov A , Funke J , Moreno - Noguier F . Accurate and Linear Time Pose Estimation from Points and Lines [A]. European Conference on Computer Vision [C]. Springer, Cham, 2016.

[14] Moreno F A , José - Luis Blanco , Javier González - Jiménez. ERODE: An Efficient and Robust Outlier Detector and its Application to Stereovisual Odometry [A]. IEEE International

Conference on Robotics & Automation [C]. IEEE, 2013.

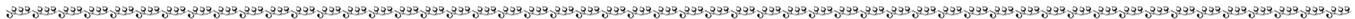
[15] GálvezLópez, Dorian, Tardós, Juan D. Bags of binary words for fast place recognition in image sequences [J]. IEEE Transactions on Robotics, 2012, 28 (5): 1188 - 1197.

[16] Arthur D , Arthur D , Vassilvtskii S , et al. k-means++: The advantages of careful seeding [A]. Eighteenth Acm - SIAM Symposium on Discrete Algorithms [C]. Society for Industrial and Applied Mathematics, 2007.

[17] Uijlings J R R , Duta I C , Sangineto E , et al. Video classification with Densely extracted HOG/HOF/MBH features: an evaluation of the accuracy/computational efficiency trade-off [J]. International Journal of Multimedia Information Retrieval, 2015, 4 (1): 33 - 44.

[18] Burri M , Nikolic J , Gohl P , et al. The EuRoC micro aerial vehicle datasets [J]. The International Journal of Robotics Research, 2016: 0278364915620033.

[19] Sturm J , Engelhard N , Endres F , et al. A Benchmark for the Evaluation of RGB - D SLAM Systems [A]. IEEE/RSJ International Conference on Intelligent Robots & Systems [C]. IEEE, 2012.



(上接第 155 页)

表 10 图形生成平台与第三方工具对比

开发方法	人员	汉字	显示符号	可变图形	软件复用率
GLStudio	美工、程序员	贴图	贴图	自主开发；显示效果差；	代码复用率低；软件重构率低；
SimApp (图形生成平台)	程序员	矢量文字	矢量图形	图形库提供；反走样处理；锯齿处理；	代码复用率高；软件重构率高；

工具进行开发的过程，该方法采用抽象工厂模式建立机载显示系统图形模型库，基于 OpenGL 技术建立机载显示系统平台，可以快速的实现飞机座舱显示系统的设计及开发，并且不受任何平台和工具环境限制，既适合实装的预研设计、开发，也适合各种平台的模拟训练系统的研制。其特点是，对开发人员的编程能力要求较低，显示软件的复用率和重构率高、继承性好，易于移植和嵌入其他第三方软件平台。利用上述平台架构和图形模型库实现的某战机座舱显示系统已经成功应用于该型飞机分队战术模拟训练系统，通过仿真试验及结果的定性定量评价分析，该模拟器完全满足训练和教学需求。

参考文献:

[1] 王和平, 王 宁, 张联梅. 综合航电开放式软件设计技术 [J]. 计算机工程与设计, 2009, 30 (1): 4 - 8.

[2] 褚文奎, 张凤鸣, 樊晓光. 综合模块化航空电子系统软件体系结构综述 [J]. 航空学报, 2009, 30 (10): 1912 - 1917.

[3] 郑凯文. 综合飞行显示系统图形生成和视频处理算法研究与 FPGA 实现 [D]. 南京: 南京航空航天大学, 2016.

[4] 高 颖, 邵亚楠, 郑 涛, 等. GL Studio 在飞行模拟器中的仿真研究 [J]. 弹箭与制导学报, 2008, 28 (1): 257 - 260.

[5] 赵双双, 孙旭东. 基于 Idata 的先进座舱多功能显示器的仿真研究 [J]. 飞机设计, 2011, 31 (3): 55 - 57.

[6] 杨卫风, 胡小琴, 蔡志勇, 等. 基于 VAPS 设计的多功能显示器仿真 [J]. 计算机应用与软件, 2011, 28 (6): 228 - 230.

[7] 华铨平, 庞倩超, 谢 颖. 抽象工厂设计模式在 3 层结构开发中的应用 [J]. 大庆石油学院学报, 2009, 33 (3): 112 - 115.

[8] Brian Salomon, Christopher Dalton, Allan MacKinnon. Path rendering by counting pixel coverage [J]. Technical Disclosure Commons, 2017, 5, 17.

[9] 李晓伟, 徐冰霖, 张银发, 等. 设计模式在测控通信构件设计中的应用 [J]. 飞行器测控学报, 2012, 31 (6): 63 - 67.

[10] 王志乐, 董军宇. 航空装备统一建模与仿真训练平台研究 [J]. 计算机测量与控制, 2018, 26 (8): 127 - 132.

[11] 姜丹丹, 李成贵. 基于 ARM 与 FPGA 的主飞行仪表显示系统的设计 [J]. 计量与测试技术, 2009, 36 (1).

[12] Cyril Crassin, Morgan McGuire, Kayvon Fatahalian, Aaron Lefohn. Aggregate G-buffer anti-aliasing [A]. i3D'15 Proceedings of the 19th Symposium on Interactive 3D Graphics and Games [A]. San Francisco, California, 2015, 2, 27, 109 - 119.

[13] Eric Enderton, Eric Lum, Christian Rouet, Oleg Kuznetsov. Accumulative anti-aliasing [A]. SIGGRAPH '15 Special Interest Group on Computer Graphics and Interactive Techniques Conference [C]. Los Angeles, California, 2015, 8, 9.

[14] 高艳辉, 肖前贵, 胡寿松, 等. 飞行模拟器发展中的关键技术 [J]. 计算机测量与控制, 2014, 22 (2): 587 - 560.