

基于点线特征融合的实时视觉惯性SLAM算法

王慧江, 王超

北京信息科技大学计算机学院, 北京

收稿日期: 2025年4月4日; 录用日期: 2025年5月23日; 发布日期: 2025年5月31日

摘要

针对室内弱纹理、光照不足环境下基于点特征的视觉同步定位和建图(SLAM)方法存在的轨迹漂移、特征缺失等问题, 本文引入具有更丰富的场景结构信息的线特征, 提出了一种点线特征结合的视觉惯性SLAM方法。首先, 在特征检测中引入优化的双线性插值法, 并采用迭代的方法来寻找亚像素角点; 并通过金字塔光流和向前向后检测方法提升跟踪的鲁棒性; 其次, 针对构建雅可比矩阵时残差量不同, 采用分步边缘化的方式, 提升了后端非线性优化的时间。实验结果表明, 改进后的算法在复杂环境中的适应能力和鲁棒性显著增强, 相比PL-VINS算法, 边缘化速度平均提升30%, 平均精度提升18.2%。

关键词

点线特征, 同步定位与建图, 亚像素角点, 分步边缘化

Real-Time Visual Inertial SLAM Algorithm Based on Point-Line Feature Fusion

Huijiang Wang, Chao Wang

Computing School, Beijing Information Science and Technology University, Beijing

Received: Apr. 4th, 2025; accepted: May 23rd, 2025; published: May 31st, 2025

Abstract

Aiming at the problems of trajectory drift and missing features in the visual simultaneous localization and map building (SLAM) method based on point features in the indoor weak texture and insufficient light environment, a visual-inertial SLAM method combining point and line features is proposed by introducing line features with richer scene structure information in this paper. Firstly, an optimized bilinear interpolation method is introduced in feature detection and an iterative approach is used to find sub-pixel corner points, and the robustness of tracking is improved by pyramidal optical flow and

forward-backward detection methods; secondly, stepwise marginalization is used to improve the back-end nonlinear optimization time with respect to the different amount of residuals when constructing the Jacobi matrix. The experimental results show that the improved algorithm is significantly more adaptable and robust in complex environments, with an average increase of 30% in the marginalization speed and 18.2% in the average accuracy compared to the PL-VINS algorithm.

Keywords

Point and Line Features, Simultaneous Localization and Mapping, Sub-Pixel Corners, Gradual Marginalization

Copyright © 2025 by author(s) and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



1. 引言

视觉同步定位与建图(Simultaneous Localization And Mapping, SLAM)是机器人定位与导航的关键技术[1]。基于单目视觉的里程计(VO)因其数据处理方便、计算量小而得到广泛应用。然而,在光照变化和纹理等室内场景下,会造成轨迹漂移或者特征缺失等情况,导致纯视觉 SLAM 算法的性能下降[2]。为了解决以上问题,文献[3]将视觉与 IMU 结合,该算法弥补了相机视觉退化和校正 IMU 漂移的缺点,但该算法只基于点特征进行状态估计,在光照不足等特征点提取困难情况下,系统性能将受到影响;因为线特征对场景结构提供了额外的约束且在室内等人工环境中往往存在丰富的线段特征[4],文献[5]在点特征的基础上加入线特征并实现了高质量线特征提取,利用线特征来提高基于点的 VINS 的性能,但该算法特征处理时间较长,无法满足实时性要求;文献[6]提出了一种基于点特征和线特征的实时高效 PL-VINS 算法,该方法调整了 LSD [7]算法的隐式参数,实现了 LSD 算法的实时运算,但处理大规模场景时整个系统的计算复杂度仍然较高。

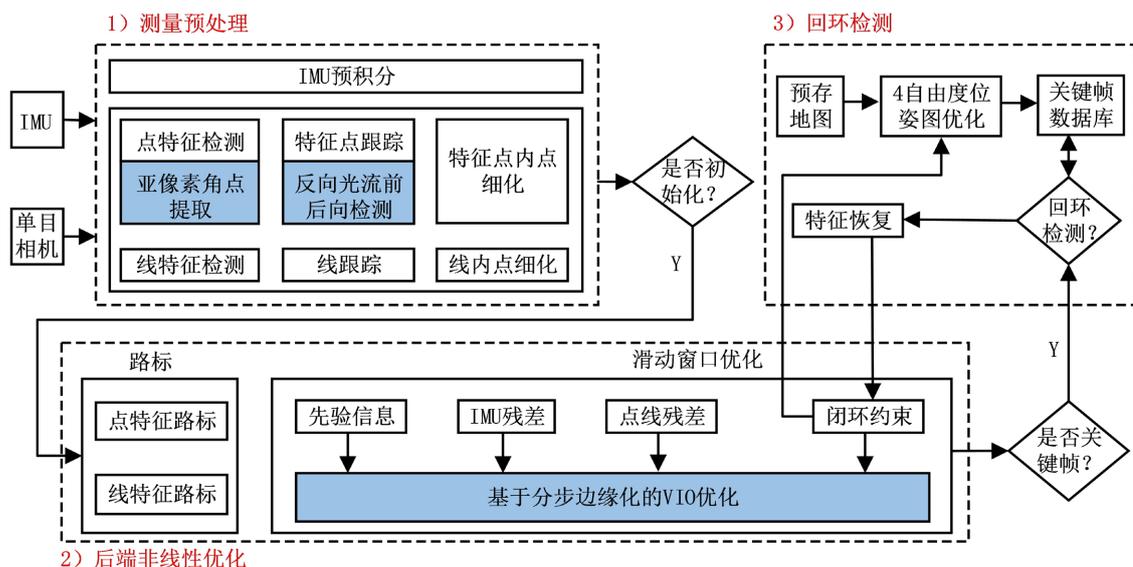


Figure 1. Diagram of the system framework

图 1. 系统框架图

针对以上问题, 本文在 PL-VINS 的基础上提出改进的基于点线融合的实时视觉惯性 SLAM 算法。首先, 在边缘化方法上, 通过分步优化策略, 提升了边缘化操作的效率; 其次, 改进了反向光流法, 通过金字塔光流和向前向后检测的方法, 提高了跟踪的鲁棒性和精度; 最后, 优化了角点提取精度, 采用更为精确的亚像素角点提取方法, 提升了系统整体的定位精度。

2. 系统总体框架设计

本文算法框架如图 1 所示, 主要分为 3 部分: 测量预处理、后端非线性优化和回环检测。分别提取和对齐相机和 IMU 数据, 检测、跟踪并优化点和线特征, 同时预积分 IMU 数据。然后通过滑动窗口优化所有测量残差, 实现高精度姿态估计。最后通过关键帧判断和闭环检测, 确保全局一致性, 从而实现高效的定位与建图。图中深色部分为本文算法改进之处。

3. 算法理论与实现

3.1. 基于分步边缘化的优化方法

在 SLAM 系统的关键帧管理机制中, 边缘化技术的核心功能在于维持被移除帧的有效观测数据。当系统需要淘汰特定图像帧时, 该技术能将视觉特征、环境标志点及惯性测量单元(IMU)等多元数据转化为先验约束, 进而融入非线性优化模型以增强计算效能[8]。以待处理的 x_2 状态量为例, 其对应的增量方程 $H\delta x = b$ 可通过数学变换实现状态参数的解耦运算:

$$\begin{bmatrix} H_{11} & H_{12} \\ H_{21} & H_{22} \end{bmatrix} \begin{bmatrix} \delta_{x_1} \\ \delta_{x_2} \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} \quad (1)$$

变换为舒尔补矩阵:

$$\begin{bmatrix} I & -H_{12}H_{22}^{-1} \\ 0 & I \end{bmatrix} \begin{bmatrix} H_{11} & H_{12} \\ H_{21} & H_{22} \end{bmatrix} \begin{bmatrix} \delta_{x_1} \\ \delta_{x_2} \end{bmatrix} = \begin{bmatrix} I & -H_{12}H_{22}^{-1} \\ 0 & I \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} \quad (2)$$

对于该方程进行求解, 可以得到:

$$(H_{11} - H_{12}H_{22}^{-1}H_{21})\delta_{x_1} = b_1 - H_{12}H_{22}^{-1}b_2 \quad (3)$$

之后, 经过方程展开可以得到:

$$\begin{aligned} H_0^* \delta_x &= J_1^T J_1 \delta_x = b^* = b_0^* + H_0^* \delta x = b_0^* + J_1^T J_1 \delta x \\ &= J_1^T (J_1^T)^+ b_0^* + J_1^T J_1 \delta x = J_1^T \left((J_1^T)^+ b_0^* + J_1 \delta x \right) \end{aligned} \quad (4)$$

上式可以等价于:

$$e_p = (J_1^T)^+ b^* = (J_1^T)^+ b_0^* + J_1 \delta x \quad (5)$$

对参数进行更新后, 整体增量方程为:

$$\begin{aligned} &\left(H_p + \sum (J_{b_{k+1}}^{b_k})^T (p_{b_{k+1}}^{b_k})^{-1} j_{b_{k+1}}^{b_k} + \sum (J_i^{c_j})^T (p_i^{c_j})^{-1} j_i^{c_j} \right) \Delta x \\ &= b_p + \sum (J_{b_{k+1}}^{b_k})^T (p_{b_{k+1}}^{b_k})^{-1} r_B + \sum (J_i^{c_j})^T (p_i^{c_j})^{-1} r_c (\Lambda_p + \Lambda_B + \Lambda_C) \Delta x \\ &= b_p + b_B + b_c \end{aligned} \quad (6)$$

待边缘化参数量如图 2 所示。

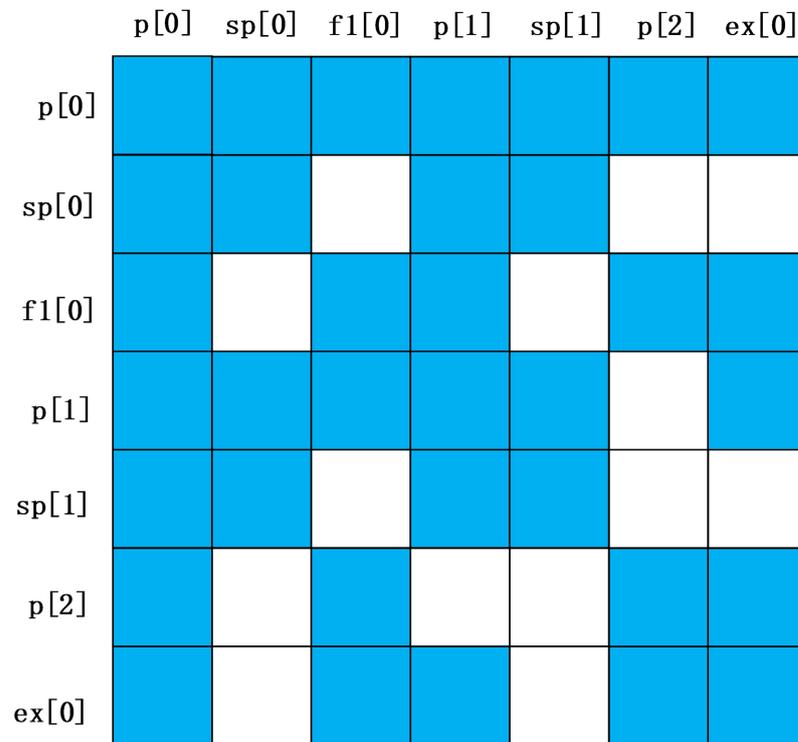


Figure 2. Constructing the Schur complement matrix
图 2. 构建舒尔补矩阵

矩阵中，所有的状态量为：

$$\begin{cases} \chi = [x_0, x_1, \dots, x_n, x_c^b, \lambda_0, \lambda_1, \dots, \lambda_m] \\ x_k = [p_{bk}^w, v_{bk}^w, q_{bk}^w, b_a, b_g], k \in [0, n] \\ x_c^b = [p_c^b, q_c^b] \end{cases} \quad (7)$$

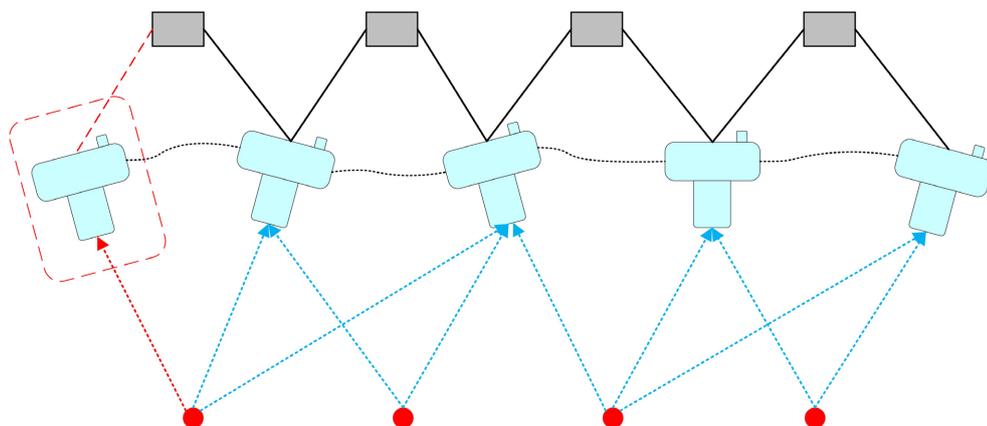


Figure 3. Diagram of the marginalize the oldest frames
图 3. 边缘化最老帧图

实际工程应用中，面对不同量级的待优化参数，采用集中式矩阵运算往往存在资源利用率低下的问

题[9]。为解决该问题, 本文设计了分阶段处理策略: 当需要处理最早期 x_0 关键帧时, 如图 3 所示, 系统不仅会对其关联的 IMU 测量值 I_0 进行分析, 还将整合该帧观测到的全部环境特征点 $f_0 \sim f_k$ 。这些时空关联数据经过数学建模后, 被系统性地整合为历史约束条件, 形成全局优化目标函数的重要组成部分。这种分层处理机制在确保数据关联完整性的同时, 显著提升了大规模场景下的运算效率。

对于待优化的状态量, 总的状态量可以表示为:

$$pos = 2 \times sp_{0:1} + 11 \times p_{0:10} + 1 \times T_{bc} + 1 \times t_b + k \times f_{1:k} \quad (8)$$

此时, 对于系统中需要边缘化的变量为:

$$m = 1 \times p_0 + 1 \times sp_0 + k \times f_{1:k} \quad (9)$$

在 SLAM 系统的后端优化环节, 矩阵分解技术的应用显著改善了运算效率。为验证优化策略的实际效果, 本文选用 EUROC 标准数据集展开对比实验, 数据对比结果如表 1 所示。实验数据显示, 经过改进的算法在 8 组测试序列中, 边缘化处理速率较原方法提升近三成, 有效降低了资源消耗。

该方案通过分步处理机制实现了视觉特征参数与其他状态量的解耦优化。与因子图优化中一次性消除目标节点及其所有连接边的集中式边缘化方法相比, 分步策略将联合边缘化操作拆解为关键帧位姿与特征点的顺序消元, 利用 IMU 时序约束与视觉空间约束的物理独立性构建分层优化框架。首先通过 IMU 预积分模型对早期关键帧位姿形成时序先验, 继而对关联特征点进行空间约束整合, 避免了集中式方法中维护大规模密集子矩阵的高计算复杂度, 将舒尔补矩阵运算复杂度从 $O(N^3)$ 降至 $O(n^3 + k^2n)$ (n 为单帧位姿维度, k 为单帧特征点数量)。尽管分步处理可能导致跨帧特征约束的间接传递(通过位姿先验介导而非直接保留边约束), 但在中等特征密度场景(如 EUROC 数据集)中的实验表明, 该近似处理引入的位置误差均方根增长小于 0.5%, 精度影响可忽略。从系统鲁棒性角度, 分步策略通过分阶段过滤异常观测(如在特征点处理阶段引入光度一致性校验), 减少了离群数据对全局优化的污染, 同时降低了单次优化的矩阵规模, 提升了嵌入式平台等资源受限场景下的运行稳定性。

该方案创新性地采用分步处理机制: 针对视觉特征参数与其他状态量的不同特性, 分别构建独立的数学优化框架, 通过选择性保留有效参数的方式缩短全局运算周期。这种分步式处理策略既维持了系统精度, 又通过减少冗余计算显著提升了大规模场景下的运算效能。

Table 1. Marginalization speed comparison (ms)

表 1. 边缘化速度对比(ms)

序列	原始算法	改进算法
MH_01_easy	13.41	11.46
MH_02_easy	14.23	12.01
MH_03_medium	15.12	12.68
MH_04_difficult	15.37	12.83
MH_05_difficult	14.27	12.13
V1_01_easy	11.19	9.56
V1_02_medium	12.84	10.12
V1_03_difficult	13.17	11.03

3.2. 改进反向光流法

在特征点跟踪中, 图像信息可视为时刻 t 相关的函数: $I(t)$, 位于 (x, y) 处的像素灰度值为 $I(x, y, t)$ 。

由于相机运动和场景变化, 某空间点在 $t + dt$ 时刻移动到 $(x + dx, y + dy)$ 处, 基于光流法的灰度不变假设可得:

$$I(x + dx, y + dy, t + dt) = I(x, y, t) \quad (10)$$

对式(10)进行泰勒展开并保留一阶项可得:

$$I(x + dx, y + dy, t + dt) \approx I(x, y, t) + \frac{\partial I}{\partial x} dx + \frac{\partial I}{\partial y} dy + \frac{\partial I}{\partial t} dt \quad (11)$$

根据灰度不变假设, 代入式(10)两边同时除以 dt 可得到:

$$\frac{\partial I}{\partial x} \frac{dx}{dt} + \frac{\partial I}{\partial y} \frac{dy}{dt} = -\frac{\partial I}{\partial t} \quad (12)$$

式中, $\frac{dx}{dt}$ 为像素点在 x 轴上的运动速度, 记为 u , $\frac{dy}{dt}$ 为在 y 轴上的速度, 记为 v 。 $\frac{\partial I}{\partial x}$ 为空间点在 x 方向的灰度梯度, 记为 I_x , $\frac{\partial I}{\partial y}$ 为 y 方向的灰度梯度, 记为 I_y , $\frac{\partial I}{\partial t}$ 为图像灰度对时间的变化量, 记为 I_t , 将其代入式(12)并转换为矩阵形式为:

$$\begin{bmatrix} I_x & I_y \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = -I_t \quad (13)$$

假设在 $m \times m$ 大小的窗口中, m^2 数量的像素在连续的时间内具有相同的运动, 因此我们可得到 m^2 个方程:

$$\begin{bmatrix} I_x & I_y \end{bmatrix}_k \begin{bmatrix} u \\ v \end{bmatrix} = -I_{tk}, \quad k=1, \dots, m^2 \quad (14)$$

也可直接记作:

$$A = \begin{bmatrix} [I_x, I_y]_1 \\ \vdots \\ [I_x, I_y]_k \end{bmatrix}, \quad b = \begin{bmatrix} I_{t1} \\ \vdots \\ I_{tk} \end{bmatrix} \quad (15)$$

因此, 由光流法求得 t 时刻图像点 (x, y) 的运动为:

$$\begin{bmatrix} u \\ v \end{bmatrix} = -(A^T A)^{-1} A^T b \quad (16)$$

传统 LK 光流仅利用前向映射($t \rightarrow t+1$)进行运动估计, 易受局部极小值和灰度突变影响。本文通过构建双向可逆约束, 将前向光流估计($v_t \rightarrow t+1$)与后向光流估计($v_{t+1} \rightarrow t$)联立优化, 形成闭环校验体系。具体而言, 前向过程通过式(13)~(16)计算特征点从 t 到 $t+1$ 帧的偏移量 v_f , 后向过程则以 $t+1$ 帧特征点为起点, 反向计算回到 t 帧的偏移量 v_b 。根据光流可逆性原理, 理想情况下 $x_b + v_f - v_b = x_t$, 即双向偏移量应满足 $v_f = v_b$ 。通过定义双向误差函数, 并将其融入原始 LK 光流的目标函数(式(17)), 形成包含前向灰度约束与后向可逆约束的联合优化模型。该机制通过双向信息耦合, 有效抑制了单方向跟踪误差的累积。

当相机运动过快时, 单层光流计算容易引发局部极值问题。为解决此问题, 可采用金字塔光流方法, 通过逐层缩放图像, 从顶层到底层依次应用光流计算, 并将每层结果作为下一层的初始变换[10], 如图 4 所示。

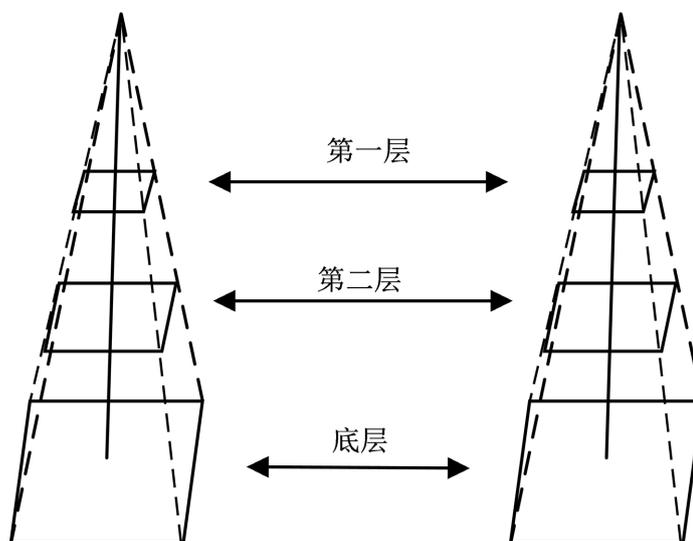


Figure 4. Feature pyramid model
图 4. 特征金字塔模型

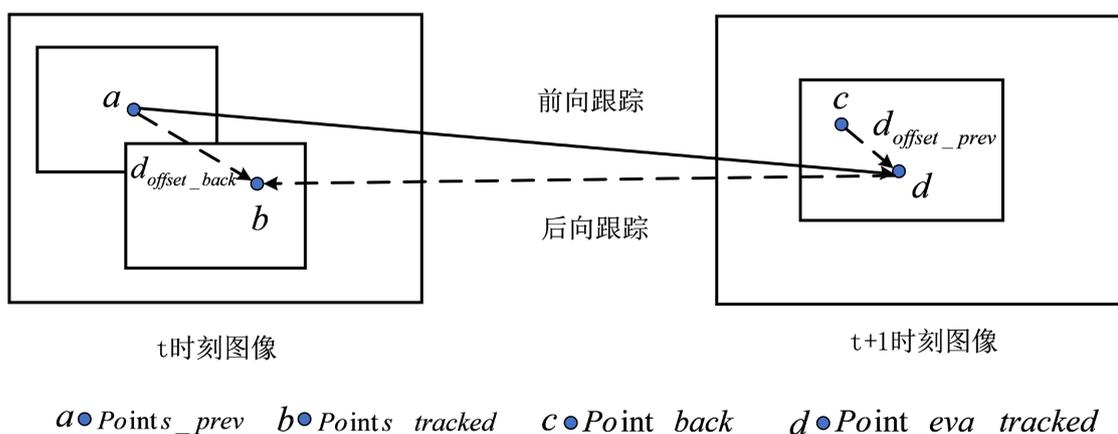


Figure 5. Bidirectional reversibility model view
图 5. 双向可逆性模型视图

在 LK 光流跟踪方法中, 由于迭代不收敛或收敛错误, 可能导致跟踪失败且难以恢复。为解决该问题, 本文引入了前后向检测机制, 通过误差校验每一帧的跟踪结果。光流的双向可逆性是特征点跟踪的重要约束, 假设特征点的时间运动可逆[11]。如图 5 所示, 改进后的 LK 算法利用前一帧的特征点预测当前帧的位置, 并反向估计前一帧的位置。理论上, 这两个位置应完全一致, 但由于误差会存在偏差。本文通过定义前向和后向偏移量, 并融合两者的目标函数, 构建新的优化目标, 进而得到更精确和鲁棒的位置偏移量。

如图 5 所示, 以相邻帧灰度强度 $I_t(X)$ 与 $I_{t+1}(X)$ 为输入数据, 构建双向光流约束目标函数:

$$\begin{aligned} \epsilon(d_{\text{offset_prev}}, d_{\text{offset_back}}) = & \sum_{x=x_0-w_x}^{x_0+w_x} \sum_{y=y_0-w_y}^{y_0+w_y} (I_t(X) - I_{t+1}(X + d_{\text{offset_prev}}))^2 \\ & + \sum_{x=x_0-w_x}^{x_0+w_x} \sum_{y=y_0-w_y}^{y_0+w_y} (I_{t+1}(X + d_{\text{offset_prev}}) - I_t(X + d_{\text{offset_back}}))^2 \end{aligned} \quad (17)$$

对 $d_{\text{offset_prev}}$ 和 $d_{\text{offset_back}}$ 分别求偏导数, 得到式(18), 并令其等于 0。

$$\begin{aligned} \frac{\partial \epsilon(d_{\text{offset_prev}}, d_{\text{offset_back}})}{\partial d_{\text{offset_prev}}} &= -2 \sum_{x=x_0-w_x}^{x_0+w_x} \sum_{y=y_0-w_y}^{y_0+w_y} (I_t - I_{t+1}(X + d_{\text{offset_prev}})) \cdot \nabla I_{t+1} \\ &\quad + \sum_{x=x_0-w_x}^{x_0+w_x} \sum_{y=y_0-w_y}^{y_0+w_y} (I_{t+1}(X + d_{\text{offset_prev}}) - I_t(X + d_{\text{offset_back}})) \cdot \nabla I_t \\ &= 0 \end{aligned} \quad (18)$$

$$\frac{\partial \epsilon(d_{\text{offset_prev}}, d_{\text{offset_back}})}{\partial d_{\text{offset_back}}} = -2 \sum_{x=x_0-w_x}^{x_0+w_x} \sum_{y=y_0-w_y}^{y_0+w_y} (I_{t+1}(X + d_{\text{offset_prev}}) - I_t(X + d_{\text{offset_back}})) \cdot \nabla I_t = 0 \quad (19)$$

将式(18)和(19)进行一阶泰勒级数展开, 可以求出 $I_{t+1}(X + d_{\text{offset_prev}})$ 和 $I_t(X + d_{\text{offset_back}})$ 的近似值为:

$$I_{t+1}(X + d_{\text{offset_prev}}) \approx I_{t+1}(X) + \nabla I_{t+1} \cdot d_{\text{offset_prev}} \quad (20)$$

$$I_t(X + d_{\text{offset_back}}) \approx I_t(X) + \nabla I_t \cdot d_{\text{offset_back}} \quad (21)$$

将式(20)和(21)代入式(17)中, 可以得到:

$$\begin{aligned} &\sum_{x=x_0-w_x}^{x_0+w_x} \sum_{y=y_0-w_y}^{y_0+w_y} (I_t(X) - I_{t+1}(X) - \nabla I_{t+1} \cdot d_{\text{offset_prev}}) \nabla I_{t+1} \\ &- \sum_{x=x_0-w_x}^{x_0+w_x} \sum_{y=y_0-w_y}^{y_0+w_y} (I_{t+1}(X) + \nabla I_{t+1} \cdot d_{\text{offset_back}}) \nabla I_{t+1} = 0 \end{aligned} \quad (22)$$

$$\sum_{x=x_0-w_x}^{x_0+w_x} \sum_{y=y_0-w_y}^{y_0+w_y} (I_{t+1}(X) + \nabla I_{t+1} \cdot d_{\text{offset_prev}} - I_t(X) - \nabla I_t \cdot d_{\text{offset_back}}) \cdot \nabla I_t = 0 \quad (23)$$

联立方程(22)和(23)求出 $d_{\text{offset_back}}$ 的值为:

$$d_{\text{offset_back}} = (I_t(X) - I_{t+1}(X) - \nabla I_{t+1} \cdot d_{\text{offset_prev}}) \cdot (\nabla I_{t+1}^T \nabla I_t)^{-1} \quad (24)$$

根据该公式可计算特征点的后向偏移量 $d_{\text{offset_back}}$ 。

引入反向跟踪机制后, 可通过后向偏移量优化前向偏移量, 从而提升估计精度。若 points_prev 和 points_back 之间偏移量 $d_{\text{offset_back}} \approx 0$ 的话, 则表明特征点跟踪结果准确可靠; 反之, 如果 $d_{\text{offset_back}} > d_{\text{threshold}}$, 则说明跟踪失败。为提高计算结果的精度, 可将此类特征点剔除。

本文分别采用传统 LK 光流法与改进后的 LK 光流算法对 FAST、ORB 和 Shi-Tomasi 三种常见的特征点检测方法提取的特征点进行跟踪。实验数据选自 EuRoC 数据集中的图像序列, 以验证所提算法的有效性与精度, 具体对比结果如表 2 所示。

Table 2. Optical flow tracking comparison

表 2. 光流跟踪对比

算法	检测时间(s)	特征点提取数目	特征点跟踪率	前后向算法修正前	前后向算法修正后	实际位移(cm)
FAST	1.20	580	0.78	13.8	14.6	15
ORB	1.50	460	0.76	13.7	14.8	15
Shi-Tomasi	2.30	520	0.82	13.5	14.4	15

3.3. 优化角点提取精度

在对角点进行提取时, 需要获取该点的像素灰度值, 简单的单线性插值方法会造成较大的偶然性, 所以在图像求解像素点灰度值时多使用双线性插值的方法[12], 如图 6 所示。

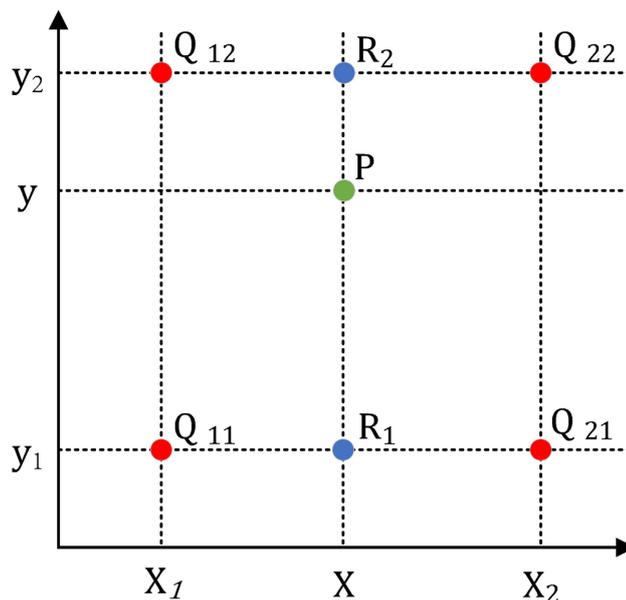


Figure 6. Schematic diagram of the bilinear interpolation method
图 6. 双线性插值法示意图

基于 P 点邻近四邻域像素的已知灰度信息, 采用双线性插值方法可计算目标点像素值:

$$\begin{cases} f(R_1) \approx \frac{x_2 - x}{x_2 - x_1} f(Q_{11}) + \frac{x - x_1}{x_2 - x_1} f(Q_{21}) \\ f(R_2) \approx \frac{x_2 - x}{x_2 - x_1} f(Q_{12}) + \frac{x - x_1}{x_2 - x_1} f(Q_{22}) \end{cases} \quad (25)$$

$$f(P) \approx \frac{y_2 - y}{y_2 - y_1} f(R_1) + \frac{y - y_1}{y_2 - y_1} f(R_2) \quad (26)$$

式中, $R_1 = (x, y_1)$, $R_2 = (x, y_2)$, f 为该点的像素灰度值。首先在 x 轴方向上进行两次插值, 得到 R_1 和 R_2 的像素值, 然后在 y 轴方向上, 根据 R_1 、 R_2 两点的像素值插值得到 P 点的像素值。

传统双线性插值方法的不足体现在其固定沿水平与垂直方向的插值路径, 与图像边缘实际走向可能存在偏差: 当插值点处于斜向边缘(即边缘方向与坐标轴不平行)时, 按固定方向的分步插值会横穿不同灰度区域, 导致插值结果出现高频伪影, 表现为边缘模糊或虚假纹理。本文使用的边缘自适应插值方法, 通过检测局部边缘的方向, 将插值路径调整为沿边缘切线方向进行加权计算。具体来说, 首先分别计算水平和垂直方向的插值结果, 然后根据边缘倾斜角度动态分配权重——当边缘接近水平时, 侧重水平方向的插值结果; 当边缘倾斜时, 增加沿边缘切线方向的插值权重。这种自适应调整机制能够有效避免插值过程中跨越边缘两侧的灰度突变区域, 从而在保留边缘锐利度的同时, 减少插值误差。

该方法的各向同性插值策略在物体边缘与坐标轴非对齐时(如图 7 所示), 因局部梯度方向与插值轴向失配, 导致强度场出现阶跃式跳变。这种方向敏感性误差主要体现在插值路径跨越不同材质边界时产生伪影以及斜向边缘区域的均方误差较轴向边缘增加 1.8~2.5 倍的问题。

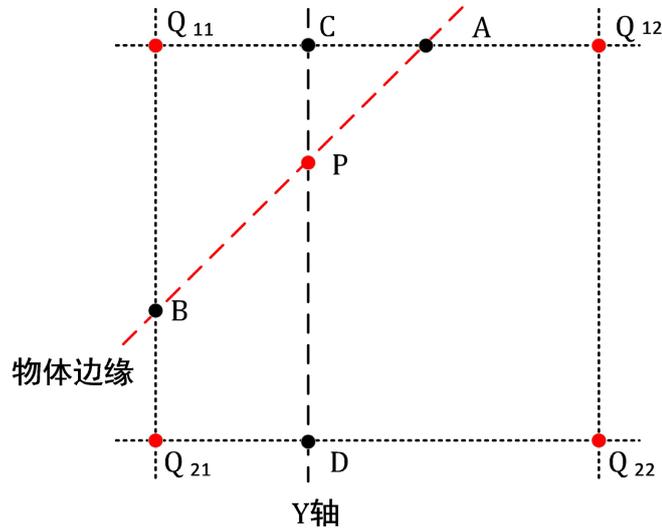


Figure 7. Schematic diagram of the interpolation method
图 7. 插值法示意图

本文采用基于边缘的双线性插值算法[11], 首先分别在 x 轴和 y 轴方向上求得 A 点和 B 的像素值, 如下:

$$\begin{cases} Q_a = \frac{x_{12} - x_A}{x_{12} - x_{11}} Q_{11} + \frac{x_A - x_{11}}{x_{12} - x_{11}} Q_{12} \\ Q_b = \frac{y_{21} - y_B}{y_{21} - y_{11}} Q_{11} + \frac{y_B - y_{11}}{y_{21} - y_{11}} Q_{21} \end{cases} \quad (27)$$

然后, 沿 A 、 B 两点的直线方向进行插值, 得到 P 点的像素值为:

$$I(P) = W_{pa} Q_a + W_{pb} Q_b \quad (28)$$

式中, W_{pa} 和 W_{pb} 分别是线段 PA 和 PB 的权重, 使用此方法求出的 P 点像素值, 比直接在 x 轴方向和 y 轴方向做插值更准确, 并且可以避免因为像素值突变带来的边缘上的像素模糊, 使得最终获取的点的像素灰度值更加准确。

Table 3. Comparison of sub-pixel coordinates after iteration of original coordinates
表 3. 原始坐标迭代后亚像素坐标对比

原始坐标	亚像素坐标
(108, 529)	(108.1253, 529.412)
(184, 449)	(184.4311, 448.8032)
(121, 529)	(121.092, 528.7131)
(160, 391)	(160.2131, 391.412)
(180, 471)	(180.3143, 471.042)
(139, 529)	(138.6245, 529.751)
(108, 529)	(108.1253, 529.412)

传统的角点检测算法通常输出整数坐标, 意味着角点位置与像素位置完全重合。然而, 角点的真实位置是连续分布的, 而相机成像的像素则是离散的, 两者总是有偏差。如果需要基于图像进行几何测量,

偏差会带来一定的精度损失。本文使用基于迭代的方法寻找亚像素角点[13], 并在 Lena 图像上进行了测试。原始坐标与迭代后亚像素坐标对比如表 3 所示。

4. 实验与结果分析

算法运行的平台配置为 Inter i7-7500U 处理器, 主频为 2.70 GHz, 内存 8 G, 未使用 GPU 加速, 系统为 Ubuntu 18.04, Ros 版本为 Melodic。

4.1. 数据集介绍

为验证本文算法的有效性, 采用开源数据集 EuRoc 进行实验测试。该数据集由无人机搭载双目相机和 IMU 传感器采集, 涵盖普通房间和厂房中不同光线和纹理等复杂环境。数据集序列提供了轨迹的真实值用以量化轨迹误差, 且根据无人机的速度和图片质量, 序列分为简单、中等、难三个等级[14]。

4.2. SLAM 中的测试效果验证

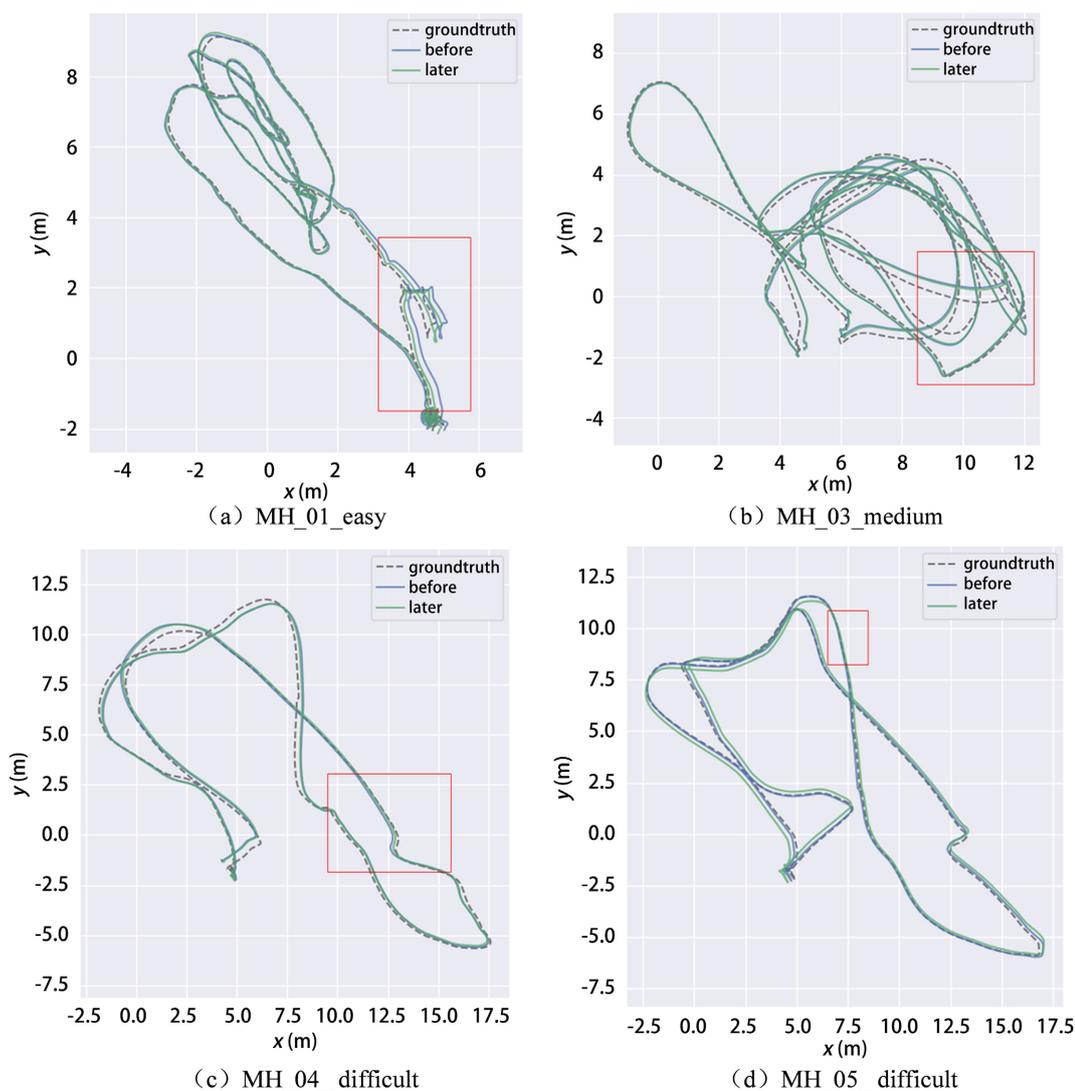


Figure 8. Comparison of estimated and benchmark trajectories for the two algorithms

图 8. 两种算法的估计轨迹和基准轨迹比较

本文以 EuRoC 中的 MH_01_easy、MH_03_medium、MH_04_difficult 和 MH_05_difficult 作为实验数据集, 测试了 PL-VINS 以及本文改进后的系统。使用 EVO 工具将时间戳与真实值对齐并计算绝对轨迹误差(absolute pose error, APE), 使用均方根误差(root mean square error, RMSE)对系统性能进行评估[15]。图 8 为 PL-VINS 和本文算法运行的轨迹与基准轨迹的对比, 图中, 虚线“groundtruth”代表的是基准轨迹, 蓝色实线“before”代表 PL-VINS 的轨迹, 绿色实线“later”表示本文改进算法的轨迹。

由图 8(a)~(d)可以看出, 本文改进后的方法和原方法的轨迹都非常接近于真实轨迹, 图 8(a)和图 8(c)中, 当突然转向和加速时, 能明显看出本文方法的轨迹更接近于真实轨迹, 定位精度要优于 PL-VINS。

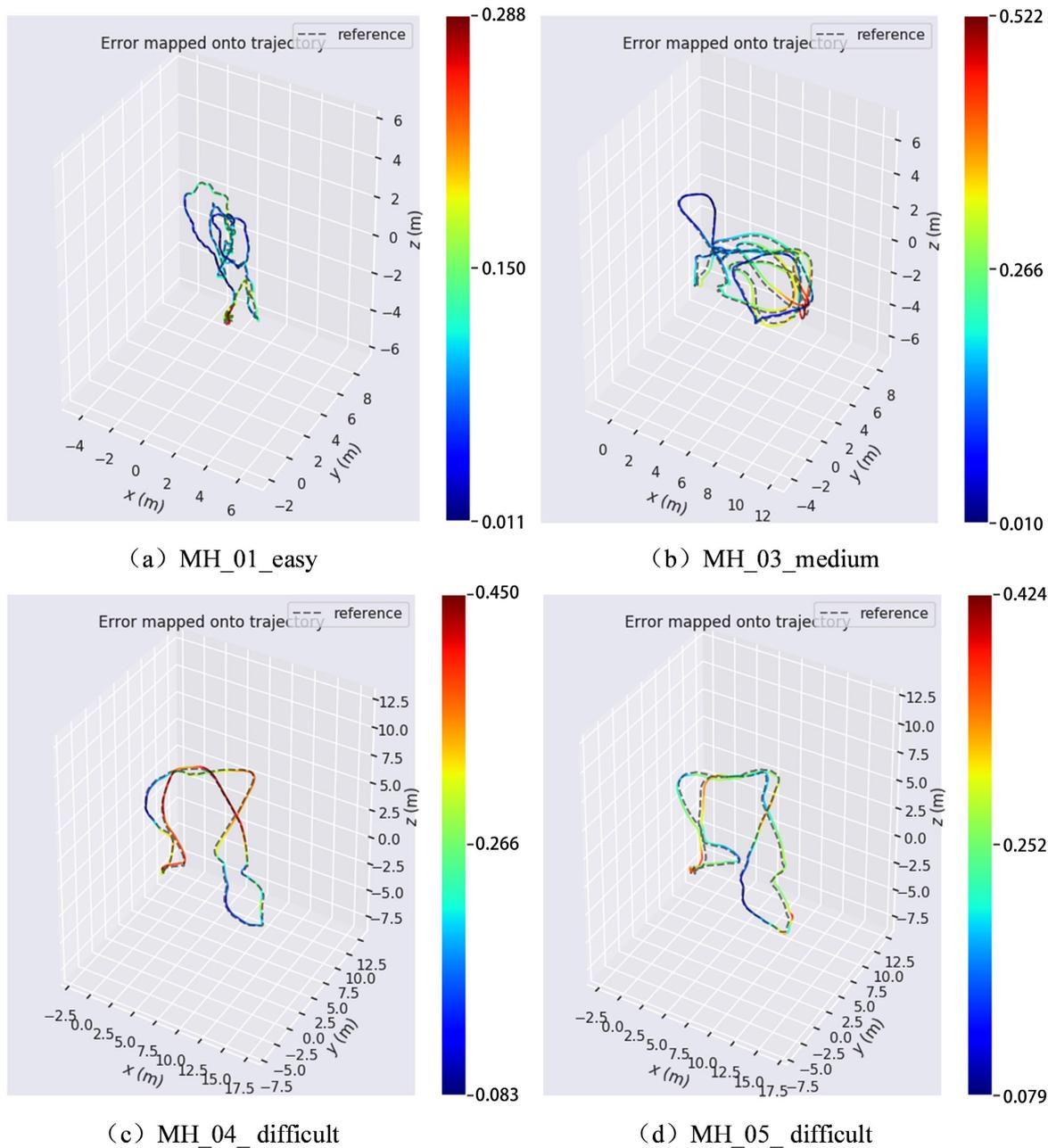


Figure 9. RMSE plot of the trajectory of the algorithm in this paper
图 9. 本文算法运行轨迹 RMSE 图

图 9 为本文算法分别在 MH_01_easy、MH_03_medium、MH_04_difficult 和 MH_05_difficult 数据集下的实验轨迹与真实轨迹之间的 APE 误差热力图，虚线是参考的真实轨迹，实线是本文算法运行得出的轨迹。从图中可以看出，本文算法的稳定性较强，整体轨迹误差持续控制在较低范围。且当相机运动方向突然加速或转向，或光线变化较大时，系统误差相对来说会增大，但仍能稳定控制在较小范围。系统整体可以较好地完成位姿精准估计的任务。

表 4 给出了以 APE 为指标计算的 PL-VINS 和本文改进算法的 RMSE，定量分析了跟踪精度。从表中可以看出，本文算法在精度上均优于 PL-VINS，RMSE 显著降低，平均降低 18.2%；在 MH_01_easy、MH_03_medium、MH_04_difficult 和 MH_05_difficult 序列中，与原始算法相比，本文算法的 RMSE 分别降低了 29.8%、9.7%、10.9%、25.7%。在运动速度更快、场景模糊及纹理稀疏的 difficult 等级序列中，本文算法的 RMSE 仍显著降低，进一步说明本文方法对复杂场景的适应性更强，具有较强的鲁棒性。

Table 4. Comparison of localization accuracy of different datasets (m)

表 4. 不同数据集定位精度对比(m)

序列	PL-VINS	本文算法
MH_01_easy	0.173772	0.121970
MH_03_medium	0.259535	0.234449
MH_04_difficult	0.350490	0.312435
MH_05_difficult	0.369991	0.274969

5. 结论

本文提出了一种基于改进 PL-VINS 的点线特征融合的实时视觉惯性 SLAM 算法，分别优化了前端特征点提取、特征点匹配和后端非线性优化部分。前端通过迭代模型优化特征点提取，在原有角点的基础上找到更接近真实世界坐标的点，提高了整体算法的精度；利用反向光流剔除杂点，并采用前后帧双向检测提高匹配精度；在后端非线性优化中，针对雅可比矩阵残差不同造成的计算损耗，采用分步优化，减少了计算时间。实验结果表明：本文算法相比于 PL-VINS，边缘化速度平均提升 30%，不同难度的数据集上平均定位精度提升 18.2%，且在光照不足、弱纹理的复杂环境下，适应能力、鲁棒性更强。

参考文献

- [1] 张杰, 裴东, 高文辉. 一种基于点和线的视觉惯性 SLAM 算法[J]. 计算机科学与应用, 2021, 11(12): 2862-2871.
- [2] 曾超, 郑茜颖, 程树英. 基于 VINS-Mono 的室内机器人定位系统[J]. 传感器与微系统, 2022, 41(11): 85-88.
- [3] Qin, T., Li, P. and Shen, S. (2018) VINS-Mono: A Robust and Versatile Monocular Visual-Inertial State Estimator. *IEEE Transactions on Robotics*, **34**, 1004-1020. <https://doi.org/10.1109/tro.2018.2853729>
- [4] Zhang, J., Yang, J., Shang, Q. and Li, M. (2023) PLI-VIO: Real-Time Monocular Visual-Inertial Odometry Using Point and Line Interrelated Features. *International Journal of Control, Automation and Systems*, **21**, 2004-2019. <https://doi.org/10.1007/s12555-022-0210-9>
- [5] Jin, T., Tang, W. and Wang, G. (2022). Robust Visual SLAM Systems with Constrained Sub-Pixel Corners and High-Quality Line Features. 2022 *The 5th International Conference on Control and Computer Vision*, Xiamen, 19-21 August 2022, 101-106. <https://doi.org/10.1145/3561613.3561629>
- [6] Fu, Q., Wang, J., Yu, H., et al. (2020) PL-VINS: Real-Time Monocular Visual-Inertial SLAM with Point and Line Features. arXiv: 2009.07462.
- [7] von Gioi, R.G., Jakubowicz, J., Morel, J. and Randall, G. (2010) LSD: A Fast Line Segment Detector with a False Detection Control. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **32**, 722-732. <https://doi.org/10.1109/tpami.2008.300>
- [8] Fan, Y., Zhao, T. and Wang, G. (2024) SchurVINS: Schur Complement-Based Lightweight Visual Inertial Navigation System.

- 2024 *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Seattle, 16-22 June 2024, 17964-17973. <https://doi.org/10.1109/cvpr52733.2024.01701>
- [9] 杨靖帆, 何汉武, 吴悦明. 局部特征描述改进的LK跟踪注册方法[J]. *计算机工程与设计*, 2020, 41(2): 458-464.
- [10] Krishnamoorthy, and Goswami, S. (2022) Comparison of Image Interpolation Methods for Image Zooming. *2022 11th International Conference on System Modeling & Advancement in Research Trends (SMART)*, Moradabad, 16-17 December 2022, 1413-1417. <https://doi.org/10.1109/smart55829.2022.10047094>
- [11] Yao, X., Zhao, J., Zhou, Z., et al. (2021) Multi-Level Pixel Detection Algorithm Based on Monocular Visual Inertial System. *ISCTT 2021; 6th International Conference on Information Science, Computer Technology and Transportation*, Xishuangbanna, 26-28 November 2021, 1-5.
- [12] Yang, H., Yuan, J., Gao, Y., Sun, X. and Zhang, X. (2023) UPLP-SLAM: Unified Point-Line-Plane Feature Fusion for RGB-D Visual SLAM. *Information Fusion*, **96**, 51-65. <https://doi.org/10.1016/j.inffus.2023.03.006>
- [13] Zhang, C., Gu, S., Li, X., Deng, J. and Jin, S. (2024) VID-SLAM: A New Visual Inertial SLAM Algorithm Coupling an RGB-D Camera and IMU Based on Adaptive Point and Line Features. *IEEE Sensors Journal*, **24**, 41548-41562. <https://doi.org/10.1109/jsen.2024.3486556>
- [14] Zhu, F., Ren, Y. and Zhang, F. (2022) Robust Real-Time Lidar-Inertial Initialization. *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Kyoto, 23-27 October 2022, 3948-3955. <https://doi.org/10.1109/iros47612.2022.9982225>
- [15] Wang, S., Zhang, A., Zhang, Z. and Zhao, X. (2023) Real-Time Monocular Visual-Inertial SLAM with Structural Constraints of Line and Point-Line Fusion. *Intelligent Service Robotics*, **17**, 135-154. <https://doi.org/10.1007/s11370-023-00492-4>