GICP?

四月十号晚上讨论

特征点
响应值？

A Real-Time Stereo Visual-Inertial SLAM System Based on Point-and-Line Features

Xin Liu, Shuhuan Wen, *Senior Member, IEEE*, and Hong Zhang, *Fellow, IEEE*

1.什么样的线特征是好特征？
2. LSD算法得到的线特征存在什么样的问题
3.如何实现优质线特征的筛选

4. IMU辅助光流跟踪特征点如何实现的？ 6.公式9是什么意思？
5. IMU辅助线特征预测如何实现的？ 7. 表格6和表格7 ORBSLAM3如何输出？

***Abstract*—Simultaneous localization and mapping (SLAM) is widely used in various fields, such as unmanned driving, robotics, and VR. The SLAM system with multiple landmarks is also a research hotspot currently. In this study, a real-time stereo visual-inertial SLAM system based on point-and-line features is proposed; the system is studied based on VINS-fusion. It improves the front-end process of VINS fusion. First, we proposed an IMU-assisted hierarchical grid optical flow tracking method that can more accurately and quickly track points between frames. Second, we add line features on the basis of existing point features. To match line features in real time, we only select the best line features to participate in optimization. We combine line segments by their geometric relationship to reduce line segment splitting and representation redundancy in the LSD algorithm. We further predict line features by IMU-assisted optical flow tracking to achieve high precision matching. In the back-end optimization process, we used a redundant structure to avoid the failure of stereo constraints in a highly dynamic environment. The proposed method outperforms the state-of-the-art methods (VINS-fusion and PL-VINS) on the EuRoC MAV dataset. The system also achieve good performance in a real environment.**

***Index Terms*—Stereo visual-inertial system, sensor fusion, line segment features, SLAM.**

I. Introduction

HIGHLY-accurate and real-time SLAM with limited computing resources has important applications in numerous fields such as robotics, augmented reality, virtual reality, unmanned vehicles, and unmanned aerial vehicles [1], [2]. In recent years, inertial navigation system is widely used in robot pose estimation. The combination of inertial and visual measurements is particularly useful in GPS-denied places [1]. Assisted by high frequency IMU information and rich visual information, visual-inertial navigation systems (VINS) solve the pose estimation problem in an unknown environment in a lightweight and energy-efficient way, and it has obvious advantages in complex environments and fast camera motion. To date, VINS methods have been widely used in visual-inertial SLAM and visual-inertial odometry. The most common VINS methods include monocular VINS methods [3], [4] and stereo VINS methods [5], [6], [7], [8]. A monocular VINS has smaller size, and lower computational cost than a stereo VINS method. However, it cannot estimate pure camera rotation. In addition, IMU can help monocular visual SLAM systems to obtain scale information. In contrary, stereo VINS will not cause scale ambigyity due to the existence of stereo constraints. In terms of overall robustness and accuracy, stereo VINS is significantly better than monocular VINS. Therefore, in our work, we mainly study SLAM methods based on stereo visual-inertial sensing.

Points have been the most widely used features in visual-inertial SLAM in recent years. A major innovation of ORB-SLAM [9] is that all modules of the system use the point feature called ORB (Oriented FAST and rotated BRIEF) [10], which makes the system robust. The ORB feature extraction results under different thresholds (threshold=40, 20, 10) are shown in Fig. 1(a)–(c). They show that the points are easy to aggregate in the region with complex gray. The corner features in low-texture areas sparse. This reduces the accuracy of SLAM, the amount of information for loop closure. In addition, ORB feature extraction and matching require a large amount of computation. To improve efficiency, many state-of-the-art methods use a sparse optical flow tracker as a lightweight front-end for motion image analysis [3], [4], [5], [7], [8], [11]. However, the optical flow calculation makes two assumptions based on the optical properties of moving objects: constant brightness and small motion between frames. This renders the system extremely sensitive to changes in illumination and rapid movement.

Use of point features is relatively mature. However, point features do not perform well in scenes with missing textures. In contrast, in many environments, there are abundant line features that complement the information from point features in the image. At the same time, compared with point features, line features are at a higher level and can describe an environment with more intuitive geometric information. The line features in most existing VINS are extracted by the LSD (line segment detector) [12] algorithm, and they are matched between frames by the LBD (line band descriptor) [13] algorithm. However, it is difficult to achieve real-time and reliable line segment

Manuscript received 2 August 2022; revised 2 October 2022 and 2 November 2022; accepted 30 December 2022. Date of publication 2 January 2023; date of current version 18 May 2023. This work was supported in part by the National Natural Science Foundation of China under Grant 62273296, in part by the National Natural Science Foundation of China and the Royal Society of Britain under Grant 62111530148, in part by the China Scholarship Council under Grant 201908130016, in part by the Hebei innovation capability improvement plan Project under Grant 22567619H, and in part by the Hebei Province Graduate Innovation Funding Project under Grant CXZZBS2022133. The review of this article was coordinated by Dr. Wenshuo Wang. *(Corresponding author: Shuhuan Wen.)*

Xin Liu and Shuhuan Wen are with the Department of Key Lab of Industrial Computer Control Engineering of Hebei Province, Engineering Research Center of the Ministry of Education for Intelligent Control System and Intelligent Equipment, Yanshan University, Qinhuangdao 06600, China (e-mail: liuxin98@stumail.ysu.edu.cn; swen@ysu.edu.cn).

Hong Zhang is with the Department of Electronic and Electrical Engineering, Southern University of Science and Technology, Shenzhen 518055, China (e-mail: hzhang@sustech.edu.cn).

Digital Object Identifier 10.1109/TVT.2022.3233721

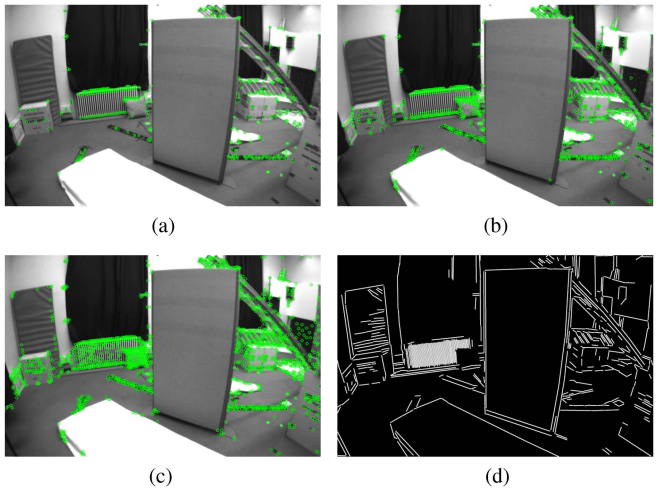


Fig. 1. Feature distribution in weak texture scenes. Direct ORB feature extraction in the indoor scene of the Euroc dataset will cause the phenomenon of feature point aggregation. And the feature points are mainly distributed in complex areas. On the contrary, the widely distributed line features in indoor low-texture scenes can make up for this deficiency. (a) Points dection (threshold=40). (b) Points dection (threshold=20). (c) Points dection (threshold=10). (d) LSD algorithm detection results.

and inter-frame line matching. As shown in Fig. 1(d), due to camera movement and unstable illumination, line segments are seriously fragmented. To overcome this situation, we propose a line feature combination method and an accurate line feature extraction and matching method.

In this work, we address the above issues in a visual-inertial SLAM system based on real-time point-and-line features detection. We propose a novel framework for the fusion of point features, line features, and IMU information in stereo visual-inertial systems. Inspired by PL-VINS [14], we use an improved LSD algorithm to extract line features quickly. The main contributions of this study are as follows:

1) We obtain uniformly distributed and stable feature points using layered grids. Simultaneously, to improve the accuracy of optical flow tracking, we integrate the IMU information for quickly calculating the prior positions of the feature points. Our feature tracking method is fast and robust and is universally applicable to all VIO/VI-SLAM systems.
2) To improve the quality of line features and remove redundant line features, we combine the nearest neighbor line features by geometric relations. In the matching process, the IMU information is used to estimate the rotation of adjacent time line features, in order to perform inter-frame feature matching more accurately.
3) We propose a stereo visual-inertial SLAM system based on optimized real-time fusion of point and line features. Point features, line features, and IMU information are integrated into our system to obtain a high accuracy of pose estimation. Simultaneously, we used point and line features to build an intuitive map in the backend.
4) We propose a fusion method for multi-vision pose information and IMU navigation information, and established redundant visual reprojection errors. Compared with existing methods, the positioning accuracy and robustness of the system are improved.

To highlight the difference between our framework and related work, we summarize the baseline-based SLAM approach relevant to our work in Table I, which makes our contribution more obvious.

II. Related Work

With the emergence of many open-source systems, visual SLAM technology has gradually matured. Some classical visual SLAM, such as ORB-SLAM2 [9], LSD-SLAM [15], and DSO [16], exhibit good performance. However, these methods still have their limitations in dealing with conditions such as fast camera motion, scenes od low texture, abrupt illimination changes and dynamic objects. In view of these limitations sensor fusion has gradually become popular and, in particular, visual odometer and IMU fusion has become a research hotspot.

Most visual-inertial SLAM methods are based on point features. The latest research ORB-SLAM3 [17] adopts the ORB feature for real-time motion estimation. However, these feature points need to be matched by calculating descriptors, which require a significant amount of computation. To reduce the computational workload, the KLT [20] sparse optical flow is used to achieve feature point matching for VINS-Mono [3] and VINS-Fusion [5]. VINS based on optical flow is widely used in unmanned driving systems because of its lightweight calculation and real-time performance [2]. To achieve faster, more robust, and accurate optical flow tracking, Xie et al. [21] proposed a hierarchical quadtree feature optical flow tracking method, which can obtain a higher accuracy than the KTL tracker. However, this method requires the extraction of corners in each layer of an image pyramid. To reduce unnecessary computation, we propose a hierarchical grid feature tracking method assisted by an IMU.

At present, there are many visual odometers based on line-segment features. Gomez et al. [22] were the first to open source SLAM system that combines point and line features. Based on this work, Gomez et al. [19] contributed to a loop-closing process by adding a novel bag-of-words approach to a stereo point-line SLAM system. Zhang [23] added line features to RGBD-SLAM to obtain more accurate results for dense indoor reconstruction. However, these works are based on pure vision sensors, which have poor robustness and accuracy in complex environments. He et al. [24] proposed a visual–inertial odometry method using point-line features based on VINS-Mono. Fu et al. [14] used hidden parameter adjustments and a line-length rejection strategy to accelerate the running speed of PL-VIO. However, monocular-inertial systems have problems with initialization and robustness. At the same time, they use the line length rejection strategy to select the longer line features, leading to the loss of some line features. In the latest SLAM method based on line features, Lim et al. [25] reconstructed a structured environment through line features and vanishing points, but the line features of the map were seriously fragmented, and the scenes mostly involve in an indoor environment, which had many limitations. To speed up the matching speed of line features, Wei et al. [26]

TABLE I
A SUMMARY OF PARTIALLY BASELINE-BASED SLAM

	Muti-sensor	Using point features	Using line features	Optical flow accelerates matching	Redundant back-end optimization
VINS-fusion [5]	✓	✓		✓	
ORB-SLAM3 [17]	✓	✓			
OpenVINS [18]	✓	✓		✓	
PL-SLAM [19]		✓	✓		
The Proposed	✓	✓	✓	✓	✓

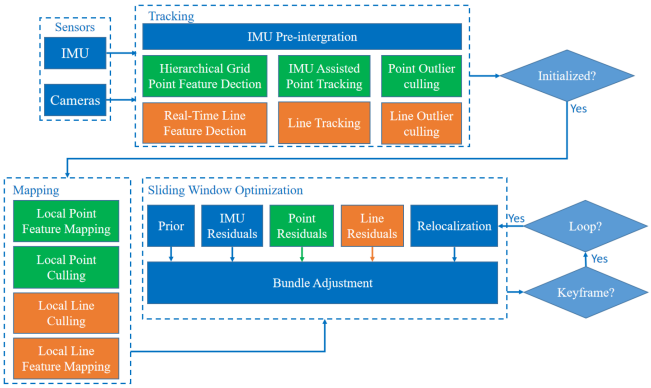


Fig. 2. The pipeline of the system.

proposed the method of matching line features by geometric distance, which greatly accelerated the matching speed. However, the matching method based on geometric distance is only applicable to the matching between adjacent frames, which is not emenable to the loop-closure detection. To improve data association between line features, Zuo et al. [27] performed robust estimation by fusing broken line features. In their recent work, Lee et al. [28] included point, line and surface features for convenient and fast tracking, and used parallel residual optimization to reduce the computational burden, with limited success. Motivated from previous research, in this study, we propose a line feature combination method to obtain longer line features, and we useIMU information and optical flow detection method in line feature matching. Different from the method of merging line features in [27], we only merge adjacent features through geometric constraints.

III. Overview and Notations

The pipeline of the proposed stereo visual-inertial SLAM system based on point and line features is shown in Fig. 2. In this system, we use a stereo camera and an IMU as sensors to obtain the measurement information. In the tracking procedure (Section IV), point and line features are extracted, matched, and screened. Meanwhile, an IMU information between two consecutive frames in the time series is pre-integrated. The optimization procedure (Section V) completes the joint optimization of the marginal prior information, IMU residual sum of squares (RSS), point feature reprojection RSS, line features reprojection RSS, and relocalization RSS.

Throughout our study, we use $(\cdot)_W$ and $(\cdot)_B$ to denote the world frame and body frame, respectively. Similarly $(\cdot)_C$ denotes the camera frame. The pose of the IMU in the world frame is defined by $[\mathrm{R}_B^W, \mathrm{p}_B^W]$, where R is the rotation matrix and p is the translation vector. g_W represents the gravity vector in the world frame. $\tilde{(\cdot)}$ represents the actual measured value with noise.

IV. Visual Inertia Processing Frontend

A. IMU Preintegration

As the basic measurement unit of inertial navigation, the IMU comprises an accelerometer and a gyroscope. IMU errors can be divided into deterministic and random errors. Deterministic errors include deterministic bias, nonlinearity, and coupling coefficient. Random errors include bias, white noise, and bias instability. We used the IMU_utils tool [29] to calibrate the IMU to compensate for its deterministic error. To estimate the IMU random error in the backend, the measurement of the IMU at time t is given by [30]

$$\begin{aligned}\tilde{\omega}_B(t) &= \omega_B(t) + \mathrm{b}^g(t) + \eta^g(t)\\ \tilde{a}_B(t) &= \mathrm{R}_B^W(a_W(t) - g_W) + \mathrm{b}^a(t) + \eta^a(t)\end{aligned} \tag{1}$$

where $\tilde{\omega}_B$ and $\tilde{a}_B$ represent the gyroscope and accelerometer measurements, respectively. ω_B and a_B are the corresponding actual values. b^g is the gyroscope bias and b^a is the acceleration bias. η^g and η^a are white noise that obeys the Gaussian distribution. Integrating the inertial measurement between the continuous kth frame and the $(k+1)$th frame, we can obtain

$$\begin{aligned}\alpha_{k+1}^k &= \int\int_{[t_k,t_{k+1}]} \mathrm{R}_W^{B_k}(t)\left[\tilde{a}(t) - \mathrm{b}^a(t)\right]dt^2\\ \beta_{k+1}^k &= \int_{[t_k,t_{k+1}]} \mathrm{R}_W^{B_k}(t)\left[\tilde{a}(t) - \mathrm{b}^a(t)\right]dt\\ \gamma_{k+1}^k &= \int_{[t_k,t_{k+1}]} \frac{1}{2}\Omega\left[\tilde{\omega}(t) - \mathrm{b}^g(t)\right]\gamma_t^{B_k}dt\end{aligned} \tag{2}$$

where α_{k+1}^k, β_{k+1}^k and γ_{k+1}^k are the preintegration value betwwen the kth frame and the $(k+1)$th frame, $\Omega(\omega) = \begin{bmatrix} -\lfloor\omega\rfloor_\times & \omega \\ -\omega^{\mathrm{T}} & 0 \end{bmatrix}$, $\lfloor\omega\rfloor_\times = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix}$. In the proposed method, we assume that the change of bias is small, so the method in VINS-Mono [3] can be used to perform a first-order Taylor approximation of α, β, and γ.

B. Feature Extraction and Matching

Usually, for two consecutive frames, the LK [31] sparse optical flow algorithm is used to complete the inter-frame matching. For long-term tracking, traditional optical flow tracking methods have difficulty handling scenes, such as illumination changes

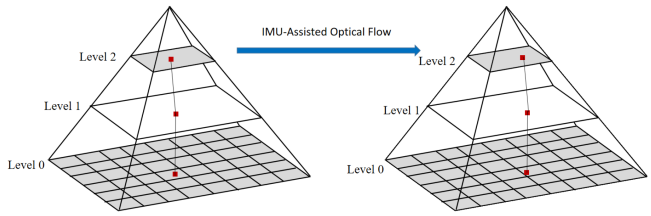


Fig. 3. IMU-Assisted Hierarchical Grid Optical Flow Tracker.

and rapid camera movement. Therefore, based on the traditional optical flow tracker, an IMU-assisted layered-grid optical flow tracking method is proposed.

For the tracking of features between frames, we used the hierarchical optical flow humanoid method. A three-layer pyramid was used to track the previous and subsequent frames. We divide the grid of the 0th level of the image pyramid (original image), sort the feature points in each grid according to the corresponding feature points, and select the feature points with a good response. This method is suitable for motion estimation during fast motion. Because the length and width of the upper levels of the image pyramid are scaled by a factor $1/2$, the displacement between pixels is also scaled by a factor of $1/2$. For the top-level od the image pyramid, we use the pre-integrated IMU value to predict the position of the optical flow to complete faster and more accurate optical flow tracking (refer to Fig. 3). Finally, the RANSAC algorithm is used to eliminate abnormal points in the matching process. For the tracking of feature points between stereo frames, after optical flow tracking, the known stereo camera external parameters are used as a prior to remove the external points. The details of the hierarchical IMU-assisted grid optical flow method are presented in Algorithm 1.

问题4

特征点响应值？

Algorithm 1: IMU-Assisted Hierarchical Grid Optical Flow.

Require: The current frame of the left camera: $cam0$, The current frame of the right camera: $cam1$
Build a pyramid of images
if $cam0$ is the first frame **then**
  $pts\_0 \longleftarrow$ Extract the FAST corner point from $cam0$
  $pts\_1 \longleftarrow$ Optical flow tracing is performed from $cam0$ to $cam1$
  $pts0\_inliers, pts1\_inliers \longleftarrow$ Eliminate outliers from $pts0$ and $pts1$
  $curr\_pts \longleftarrow$ Select the most responsive points in each grid
end if
$imu\_ang \longleftarrow$ Calculates the IMU pre-integration from the previous frame
$pts0 \longleftarrow$ Predicts the corresponding points on the current frame based on the imu_ang 预积分求对应特征点
$pts0\_level3 \longleftarrow$ On the basis of $pts0$, the optical flow tracking of the third layer pyramid is carried out between the current and the previous frame 第三层的匹配点
$pts0\_tracked \longleftarrow$ Feedback to the lower level of the pyramid until level 0 向下传播
$pts1\_tracked \longleftarrow$ Stereomatch based on $pts0\_tracked$
$pts0\_inliers, pts1\_inliers \longleftarrow$ Eliminate exclusion points from $pts0\_tracked$ and $pts1\_tracked$ level 0 原图像特征点
if the number of $pts0\_inliers < 150$ **then**
  Set mask
  $pts0 \longleftarrow$ Extract the FAST corner point from $cam0$ without mask
  $curr\_pts0 \longleftarrow$ Select the most responsive points in grid
end if
Ensure: The final matched points $pts0\_inliers$, $pts1\_inliers$, the final extracted points in the current frame of the left camera $curr\_pts0$

C. Line Feature Extraction and Matching

1) Real-Time Detection of Line Features: In visual SLAM, line features provide natural illumination and perspective invariance to SLAM. The traditional line segment detection algorithm uses edge detection algorithms, such as Canny [32] to detect the edge information of images, uses the Hough transform [33] to extract the line, and finally finds the end point of the line through segmentation. This method cannot be used in visual SLAM systems because of its low real-time performance and detection error in dense edges. LSD [12] can detect a line segment with direction, but it also has some shortcomings: it cannot meet the requirements of real-time SLAM under high dynamics because of occlusion and local blurring, the original line is cut into multiple segments, resulting in a large number of redundant short line features. This is not conducive to line feature matching between the frames. Therefore, we used the hidden parameter strategy in [14] to detect and match line features in real time.

使用线特征的好处

问题2

In this method, some hidden parameters, such as the fixed scale ratio r and fuzzy multiple N of the Gaussian pyramid are selected. We followed a standard setting of $r = 0.5$ and $N = 2$. We also selected the image scaling parameter $S = 0.5$, to scale each layer of images, and then used LSD to extract line features. A minimum length threshold $L_{\min}$ is chosen to reject the short-line features

$$L_{\min} = \lceil 0.125 * \min(W_I, H_I) \rceil \tag{3}$$

where W_I and H_I are the width and height of the input image, respectively.

2) Line Features Combination: In general, in VINS, longer line features are more stable and easier to be detected multiple times to ensure feature matching between frames. LSD, as a local line feature extractor, will produce the phenomenon of lines being split when lines intersect. At the same time, due to the characteristics of self-growth of local detection algorithm, long lines are often cut into multiple straight lines for reasons of occlusion and local blur. For this case, we merge overlapping and neighboring line segments by geometric information such as segment angle, midpoint position to line distance, and endpoint minimum distance. The main process of the algorithm is shown in Algorithm 2.

问题1

问题3

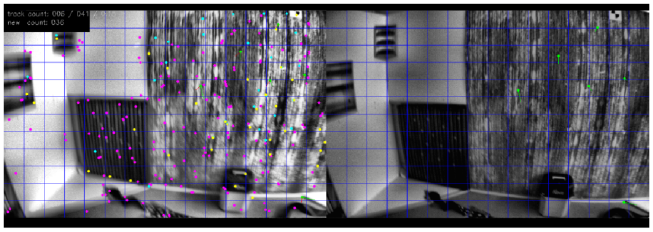


Fig. 4. Stereo matching. Among them, the green points are the stably traceable points, and the purple points are the newly added feature points through grid partitioning. Due to the constraint of dynamic response interval, there is a big difference in brightness between the right and left object images. Once the image motion blur caused by high speed motion is serious, it is difficult to find the matching point in the right eye through optical flow matching.

Algorithm 2: Line features combination.

Require: Set of line features detected in current image: $\mathrm{I} = \{I_1, I_2, I_3, \cdots I_n\}$. i, k are positive integers less than or equal to n

if $i, k \leq n$ and $i \neq j$ **then**

$\alpha_{ij} \longleftarrow$ The angle between direction vectors of I_i and I_j

$dist_{ij} \longleftarrow$ The vertical distance from the midpoint of line I_i to line I_j

$dist_{rmmin} \longleftarrow$ The minimum distance between a point in line I_i and the midpoint of I_j

end if

if $\alpha_{ij} < \alpha$ and $dist_{ij} < d$ and $mindist_{ij} < d_{min}$ **then**

$I_i \longleftarrow$ Merge I_i and I_j by gradient descent

Delete I_j

end if

Ensure: Line features set after merging line features $\mathrm{I_{com}}$

3) IMU-LK Assisted Line Feature Matching: For line feature matching between two consecutive frames, an IMU-assisted LK optical flow method is used to predict the position of line feature (refer to the section IV-B for details of IMU-LK methods). In the matching process, the IMU-LK algorithm is used to predict the position of the lines' startpoints and endpoints of the current frame. Further, we only calculate around the midpoint of the prediction line feature to find the best matching point by calculating the Hamming distance. Compared with the previous greedy search, the matching accuracy is improved and the computation is reduced.

问题5

V. Tightly Coupled Stereo VINS

As shown in Fig. 4, owing to the constraints of the camera's dynamic interval and motion blur caused by high-speed motion, it was difficult to match the right camera to the correct point by optical flow tracking in some periods. This seriously affects the positioning accuracy in complex situations. In this section, we define a new tightly coupled reprojection error of redundant structures to enhance the robustness of the system under highly dynamic and complex illumination conditions.

A. Formulation

In this study, we define the full state vector χ as

$$\begin{aligned}\chi &= [\mathrm{x}_0, \mathrm{x}_1, \ldots, \mathrm{x}_{n_k}, \mathrm{x}_C^B, \mathrm{t}_d, \lambda_0, \lambda_1, \cdots \lambda_{n_p}, \mathrm{o}_0, \mathrm{o}_1, \cdots, \mathrm{o}_{n_l}] \\ \mathrm{x}_k &= [\mathrm{p}_{B_k}^W, \mathrm{q}_{B_k}^W, \mathrm{v}_{B_k}^W, \mathrm{b}_a, \mathrm{b}_g], k \in [0, n] \\ \mathrm{x}_C^B &= [\mathrm{p}_C^B, \mathrm{q}_C^B] \end{aligned} \tag{4}$$

where x_k is the state of the IMU calculated at the time of frame k, n is the number of keyframes, p is the number of landmark points, and l is the number of 3D line segments. λ and o represent the observation of the landmark points and 3D line segments, respectively.

As shown in (1), we obtain the optimal pose estimation by minimizing the error objective function in the sliding window.

$$\min_{\chi}(\mathrm{e}_{prior} + \mathrm{e}_{imu} + \mathrm{e}_{point} + \mathrm{e}_{line} + \mathrm{e}_{loop}) \tag{5}$$

where e_{prior} and e_{loop} denote the marginal prior information and relocalization RSS, respectively. e_{imu} denotes the IMU RSS, e_{point} denotes the redundant point feature reprojection RSS, and e_{line} denotes the line feature reprojection RSS.

B. IMU Measurement Residual Model

There is an error in the measurement of the IMU at every moment, and the error satisfies the Gaussian distribution. In the recursion process of IMU pre-integration, because the value of the next moment is obtained from the value of the previous moment plus the current measurement value, the variance corresponding to the calculated value of PVQ at each moment also accumulates continuously. Therefore, it is important to determine the corresponding error transfer function.

Therefore, according to the definition of the error transfer function, we can establish an error transfer linear model of the following form, which describes the relationship between the error at the ith frame and the jth frame moment:

$$\begin{aligned}\mathrm{e}_{imu} &= \left\|\mathbf{r}_B(\tilde{\mathrm{z}}_j^i, \chi)\right\|_{\mathrm{P}_j^i}^2 \\ \mathbf{r}_{\mathrm{B}}(\tilde{\mathrm{z}}_j^i, \chi) &= \begin{bmatrix} \mathrm{R}_{\mathrm{W}}^{B_i}(\mathrm{p}_i^W - \mathrm{p}_j^W - \frac{1}{2}\mathrm{g}_{\mathrm{W}}\Delta t^2 - \mathrm{v}_i^W \Delta t) - \tilde{\alpha}_j^i \\ \mathrm{R}_{\mathrm{W}}^{B_i}(\mathrm{v}_j^W - \mathrm{g}_{\mathrm{W}}\Delta t - \mathrm{v}_i^W) - \tilde{\beta}_j^i \\ 2[q_i^{W^{-1}} \otimes q_j^W \otimes (\tilde{\gamma}_j^i)^{-1}]_{xyz} \\ \mathrm{b}_{a_j} - \mathrm{b}_{a_i} \\ \mathrm{b}_{g_j} - \mathrm{b}_{g_i} \end{bmatrix}_{xyz} \end{aligned} \tag{6}$$

where $\tilde{\mathrm{z}}_j^i$ is the IMU observation information between frames i and j. $[\cdot]_{xyz}$ extracts the vector part of the quaternion q for the error state representation. $[\tilde{\alpha}_j^i, \tilde{\beta}_j^i, \tilde{\gamma}_j^i]$ are pre-integrated IMU measurement terms using only noisy accelerometer and gyroscope measurements in the time interval between two consecutive image frames. Δt is the time difference between two consecutive frames, $\mathrm{v_i^W}$ and $\mathrm{v_i^W}$ are the camera speeds at image i and image j in the world coordinate system, respectively. B is the set of all IMU measurements.

C. Redundant Point Reprojection Residual Model

Based on the problems that the visual odometer unit may encounter, we construct a visual reprojection error model with a redundant structure. Considering the reprojection error from the 3D landmark point in the word frame $\mathrm{X_j}$ to the key point $\mathrm{x_j}$, we define the reprojection error observed in the jth image as

$$\begin{aligned} \mathrm{e}_{point} &= \mathrm{e}_s + \mathrm{e}_m \\ &= \sum_{(l,j\in P)} \rho\Big(\frac{n_m}{n_m+n_s}\left\|\mathrm{x}_m^j - \pi_m(\mathrm{R}_B^W \mathrm{X}^j + \mathrm{p}_B^W)\right\|^2_{\Sigma_l^{C_j}} \\ &\quad + \frac{n_s}{n_m+n_s}\cdot\left\|\mathrm{x}_s^j - \pi_s(\mathrm{R}_B^W \mathrm{X}^j + \mathrm{p}_B^W)\right\|^2_{\Sigma_l^{C_j}}\Big) \end{aligned} \tag{7}$$

where ρ is the Huber norm [34], x_m^j is the 2D key point in the left camera, x_s^j is the 3D key point in the stereo camera, P is the set of point features in the sliding window and Σ is the covariance matrix. n_m and n_s represent the number of feature points tracked between adjacent inter-frame and the number of feature points tracked between stereo inter-view, respectively. The projection function of stereo π_s and the projection function of monocular π_m are defined as follows:

$$\begin{aligned} \pi_m\left(\begin{bmatrix} X \\ Y \\ Z \end{bmatrix}\right) &= \begin{bmatrix} f_x\frac{X}{Z} + c_x \\ f_y\frac{X}{Z} + c_y \end{bmatrix} \\ \pi_s\left(\begin{bmatrix} X \\ Y \\ Z \end{bmatrix}\right) &= \begin{bmatrix} f_x\frac{X}{Z} + c_x \\ f_y\frac{X}{Z} + c_y \\ f_y\frac{X-b}{Z} + c_x \end{bmatrix} \end{aligned} \tag{8}$$

where $[f_x, f_y]$ is the focal length, $[c_x, c_y]$ is the principal point, and b is the baseline of the camera.

plucker coordinates?

Huber norm?

D. Line Reprojection Residual Model

We express the kth space line as the Plücker coordinates in the camera coordinate system as $L_k = \begin{bmatrix} \mathrm{n}_C \\ \mathrm{d}_C \end{bmatrix}$. The projection of the kth space line on the ith image is

$$\mathrm{l}_k = [l_1, l_2, l_3]^{\mathrm{T}} = \mathrm{K}_l \cdot \mathrm{n}_C \tag{9}$$

n表示的是直线上的一个方向向量，d表示的是直线上的一个点

where K_l denotes the line projection matrix [35].

Finally, the reprojection residual of the kth space line L_k in the jth camera is

$$\mathrm{e}_{line} = \sum_{(j,k)\in L} \left(\rho\left\|\mathrm{d}(\mathbf{m}, \mathbf{l})\right\|^2_{\Sigma_{L_k}^{C_j}}\right) \tag{10}$$

where $\mathrm{d}(\mathbf{m}, \mathbf{l})$ denotes the distance from the point to the line, and $\mathbf{m}$ is the midpoint of a line feature. ρ is the Huber norm [34].

VI. Experiments

In this section, the experimental results prove the effectiveness of the proposed algorithm. We evaluated the performance of the proposed algorithm in different lighting, texture loss, motion blur, and other environments using the public EuRoC [36] dataset as the benchmark. Simultaneously, we qualitatively demonstrated the physical robot in a real environment to evaluate the real-time performance and effect of the mapping. All experiments were performed on a computer with an Intel i7 CPU and 16 GB of RAM. The robot uses Turtlebot 2 as a mobile platform, and the image sequence was captured by a Mynt Eye stereo camera.

TABLE II
Sequences Characteristics

Sequences	Avg. Vel/Angular Vel.	Characteristics
MH_03	$0.99\mathrm{m}\cdot\mathrm{s}^{-1}/0.29\mathrm{rad}\cdot\mathrm{s}^{-1}$	fast motion, bright scene
MH_05	$0.88\mathrm{m}\cdot\mathrm{s}^{-1}/0.21\mathrm{rad}\cdot\mathrm{s}^{-1}$	fast motion, dark scene
V1_03	$0.75\mathrm{m}\cdot\mathrm{s}^{-1}/0.62\mathrm{rad}\cdot\mathrm{s}^{-1}$	fast motion, motion blur
V2_02	$0.72\mathrm{m}\cdot\mathrm{s}^{-1}/0.59\mathrm{rad}\cdot\mathrm{s}^{-1}$	fast motion, bright scene

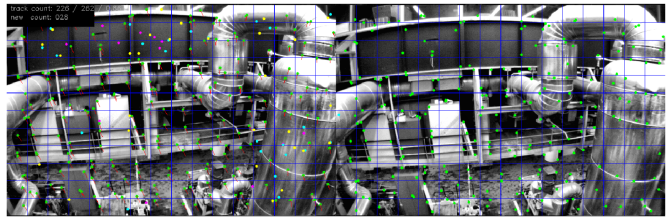


Fig. 5. Feature point detection and matching. Among them, the green points are the stably traceable points, and the purple points are the newly added feature points through grid partitioning.

A. Evaluation Using EuRoC Dataset

The EuRoC [36] dataset contains two scenarios for indoor MAV: a machine Hall and a normal room, which includes stereo images captured by a global shutter camera and IMU measurements. Each sequence provides real trajectories measured by the Leica Nova MS50 or Vicon motion capture system. We used the real trajectory measured by Leica Nova MS50 as the groundtruth.

In our experiment, MH_03_medium, MH_05_difficult, V1_03_difficult, and V2_02_medium sequences in the EuroC dataset were used to evaluate the proposed system. The characteristics of the sequences are listed in Table II. In our paper, we use absolute trajectory error (ATE) and relative pose error (RPE) to quantitatively evaluate the positioning accuracy. Where ATE represents the global consistency of positioning. In RPE, we evaluate the translation error and rotation error respectively, And the fixed-time-delta is set to 1 deg, 1 m and 1 s, respectively [37].

1) Ablation Experiment: In this part, we analyze the effect of point features and line features on the system, respectively.

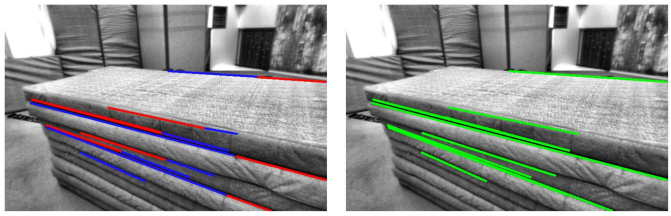
For point features, Fig. 5 shows our point feature matching process during operation, where the green points represent the key points that can be tracked and matched in the right camera steadily. Table III compares the tracking speed of the IMU-assisted hierarchical grid optical flow method and the speed of traditional LK in the fast moving continuous 20 frames of the MH_05_difficult sequence. The results show that the proposed method is lower than the traditional LK optical flow method in 17 out of 20 consecutive frames in terms of time cost. This is because the prediction of feature point positions based on IMU preintegration can provide a good position benchmark for optical flow estimation.

TABLE III
COMPARISON OF THE SPEED OF OPTICAL FLOW TRACKING

Sequence number	1	2	3	4	5	6	7	8	9	10
LK tracker cost (ms)	1.437	1.398	1.548	6.332	2.023	**2.823**	3.488	1.026	2.266	**1.464**
The proposed cost (ms)	**1.200**	**1.384**	**0.852**	**1.970**	**1.975**	3.187	**2.685**	**0.902**	**1.650**	2.429
Sequence number	11	12	13	14	15	16	17	18	19	20
LK tracker cost (ms)	2.213	1.250	4.213	1.896	**6.184**	0.878	0.751	0.883	0.900	2.486
The proposed cost (ms)	**0.989**	**1.212**	**2.477**	**1.720**	7.962	**0.650**	**0.710**	**0.655**	**0.684**	**1.087**

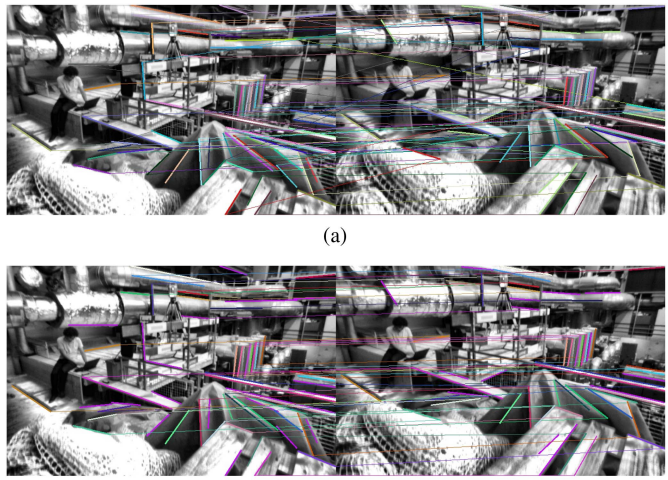
TABLE IV
COMPARISON OF THE SPEED OF LINE FEATURE TRACKING

	detect prosessing	tracker process
Mean cost (ms)	24.896	32.055



(a) (b)

Fig. 6. Example of Merging Process of Line Features on V2_02_medium. (a) Redundancy and fracture lines. (b) Line features after merging.



(a) (b)

Fig. 7. Line Feature Matching. (a) Select the global best descriptor for matching. (b) Line Feature Matching for Optical Flow Prediction Based on IMU-LK.

For the line feature, we pay attention to its real-time performance, so we separately run the line feature tracking thread in the MH_05_diffcult sequence. The average time consumed by feature detection and matching is shown in Table IV. According to the results, the improved line feature detection and matching can be run in real time on the CPU and the camera shutter frequency is 20 Hz. The result of line feature combination is shown in Fig. 6. The red and blue lines in Fig. 6(a) represent the line features of fracture caused by illumination and occlusion et al., and the green lines in Fig. 6(b) represent the line features after merging the red and blue line features. It can be seen from the figure that the method adopted by us can reduce line segment splitting caused by LSD algorithm. The merged line features can better represent the image for more stable tracking. We matched the line features between two frames by direct matching and IMU-LK assisted matching, respectively. The results are shown in Fig. 7. It is obvious that our method can better match the line features.

2) Accuracy Comparison Experiment: Figs. 8–11 show the VINS-fusion, the OpenVINS [18], the PL-VINS and the proposed method in the EuRoC dataset for the visualization of 3D trajectory comparison and ATE comparison. In Figs. 8–11, the black line represents the groundtruth of the corresponding sequence, and the blue line represents the trajectory measured by the corresponding algorithm. After we align the scales, the APE between the corresponding time stamps is drawn as red, and the length of the red line represents the size of the APE at this time. In Figs. 8 and 9, it is obvious that the length of the trajectory error line displayed by our method is shorter than that of the other methods, and in Figs. 10–11, the trajectory error line displayed by our method is similar to OpenVINS, but both are better than the other two methods. For a clearer comparison, in Fig. 12, we compared the ATE of these algorithms on EuRoC with the box diagram. The red boxes show our result. From the results, our method keeps the same effect as OpenVINS in V1_03 sequence, and outperforms other algorithms in other sequences. The results show that in most cases, the proposed algorithm can achieve higher accuracy than other algorithms in fast motion, motion blur and illumination changing scenes. 平均误差和标准差

In addition, we provide specific comparisons across all sequences in the EuRoC dataset in V, VI, and VII. We present the RMSE, mean error and standard deviation (S.D.) of ATE in this paper, while RMSE and S.D. are more relevant because they can better indicate the robustness and stability of the system. For RPE, we chose fixed time delta = 1 deg, 1 m, 1 s to calculate the RMSE of RPE, respectively. From Tables V–VII, in most cases, our method outperforms VINS-fusion, OpenVINS and PL-VINS in terms of ATE, RPE-translation and RPE-rotation. In ATE, the proposed method outperforms VINS-Fusion and OpenVINS in 8 sequences, mainly because the introduction of line features increases constraints, and IMU preintegration provides a good benchmark for matching points. In ATE, the proposed methods are superior to PL-VINS, mainly because of the introduced stereo baseline constraints and high precision

RPE: relative pose error

绝对位姿误差（ATE）：它是指估计位姿和真实位姿之间的直接差值。
相对位姿误差（RPE）：它用于计算相同两个时间戳上的位姿变化量的差。

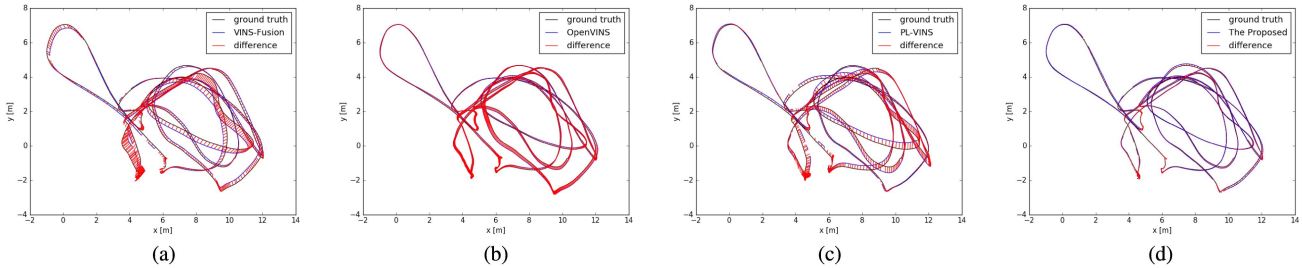


Fig. 8. Comparison of the trajectories and ATEs on MH_03_medium. (a) VINS-fusion. (b) OpenVINS. (c) PL-VINS. (d) The Proposed.

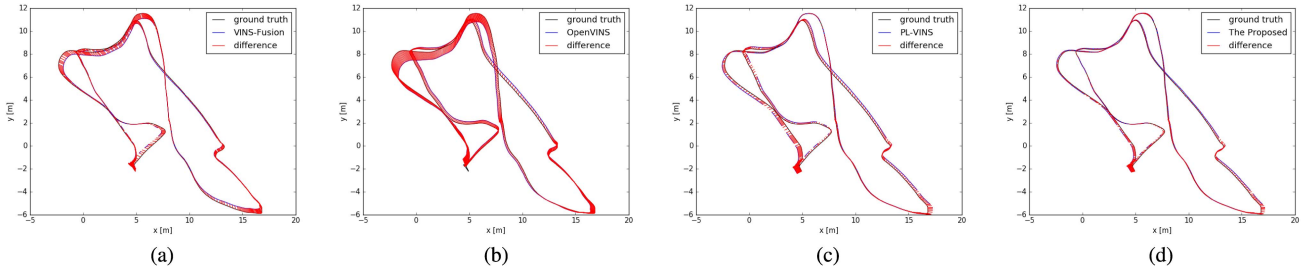


Fig. 9. Comparison of the trajectories and ATEs on MH_05_difficult. (a) VINS-fusion. (b) OpenVINS. (c) PL-VINS. (d) The Proposed.

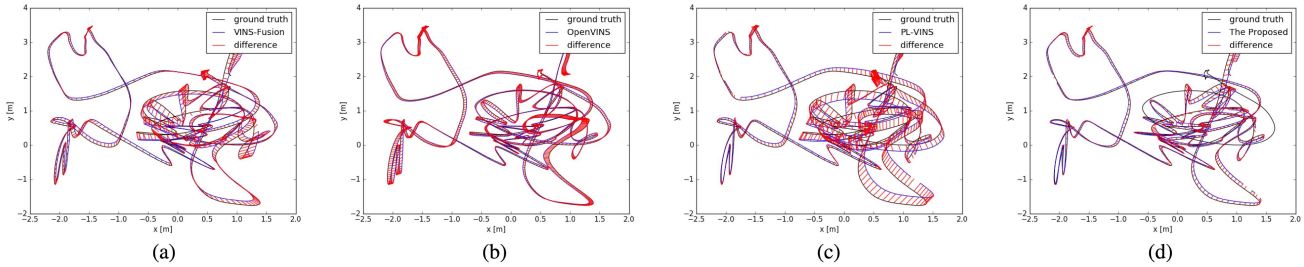


Fig. 10. Comparison of the trajectories and ATEs on V1_03_difficult. (a) VINS-fusion. (b) OpenVINS. (c) PL-VINS. (d) The Proposed.

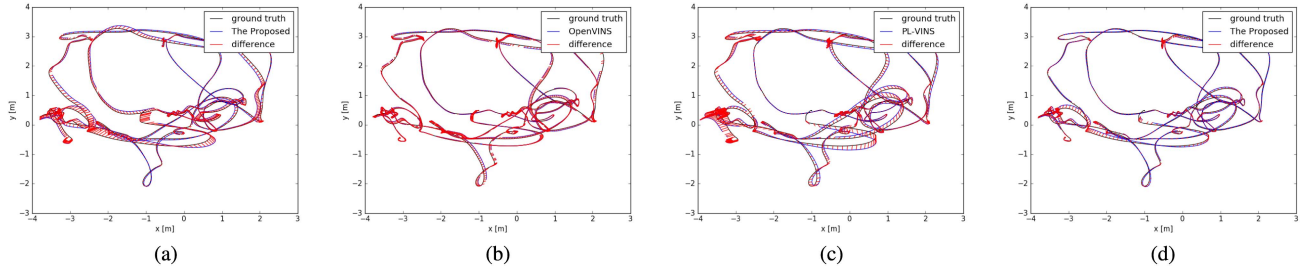


Fig. 11. Comparison of the trajectories and ATEs on V2_02_medium. (a) VINS-fusion. (b) OpenVINS. (c) PL-VINS. (d) The Proposed.

line feature matching. This means that the proposed method can perform well in most scenes in terms of global trajectory consistency and algorithm accuracy. In RPE-Translation and RPE-Rotation, among all results, VINS-Fusion got 8(8+0) best results, OpenVINS got 19(5+14) best results, PL-VINS got 17(12+5) best results, and the proposed method achieved 36(24+12) best results. This means that the drift in the proposed system is smaller than other baseline-based methods and point-line fusion methods. However, it is worth noting that OpenVINS need to be initialized statically, so it cannot run stably in MH_04. At the same time, there will be large scale errors in MH_01, MH_03 and MH_05, and the results need to be evaluated by scale alignment. It can be seen from the comparison of S.D. that our method has the highest robustness compared with VINS-Fusion, OpenVINS and PL-VINS. Obviously, the redundant residual structure can ensure the robustness of the system in fast motion, motion blur and illumination changing scenes.

TABLE V
MEAN ABSOLUTE TRAJECTORY ERROR (ATE) IN UNITS OF METERS FOR TEN RUNS OF EACH ALGORITHM ON THE EUROC

Sequences	VINS-fusion(Stereo+IMU)			OpenVINS(Stereo+IMU)			PL-VINS			OURS		
	RMSE	Mean	S.D.	RMSE	Mean	S.D.	RMSE	Mean	S.D.	RMSE	Mean	S.D.
MH_01	0.247	0.224	0.102	0.085	0.082	0.059	0.147	0.125	0.076	**0.064**	**0.060**	**0.021**
MH_02	0.199	0.181	0.082	0.139	0.128	0.036	0.161	0.123	0.104	**0.104**	**0.085**	**0.020**
MH_03	0.276	0.248	0.121	0.117	0.109	0.042	0.236	0.213	0.103	**0.069**	**0.065**	**0.029**
MH_04	0.423	0.380	0.184	/	/	/	0.304	0.281	0.114	**0.154**	**0.145**	**0.039**
MH_05	0.310	0.284	0.125	0.450	0.398	0.212	0.265	0.241	0.110	**0.181**	**0.162**	**0.081**
V1_01	0.112	0.104	0.043	0.058	0.055	0.019	0.068	0.059	0.035	**0.049**	**0.046**	**0.016**
V1_02	0.100	0.091	0.039	**0.052**	**0.045**	**0.025**	0.118	0.101	0.061	0.057	**0.045**	0.035
V1_03	0.109	0.098	0.050	**0.061**	**0.054**	**0.016**	0.174	0.154	0.081	0.068	0.062	0.028
V2_01	0.130	0.108	0.073	0.057	0.048	0.030	0.071	0.054	0.046	**0.043**	**0.037**	**0.022**
V2_02	0.108	0.091	0.075	0.053	0.048	0.021	0.114	0.010	0.058	**0.049**	**0.041**	**0.027**
V2_03	0.308	0.280	0.127	0.141	0.135	**0.041**	0.277	0.245	0.126	**0.112**	**0.102**	0.048

Note that the bold means the best, and "/" means that cannot properly. All results include loop closure.

TABLE VI
MEAN RPE-TRANSLATIONAL IN UNITS OF METERS FOR TEN RUNS OF EACH ALGORITHM ON THE EUROC

Sequences	VINS-fusion(Stereo+IMU)			OpenVINS(Stereo+IMU)			PL-VINS			OURS		
	1 deg	1 m	1 s	1 deg	1 m	1 s	1 deg	1 m	1 s	1 deg	1 m	1 s
MH_01	**0.005**	0.040	0.026	0.006	0.043	0.028	**0.005**	**0.035**	0.023	0.006	**0.035**	**0.022**
MH_02	**0.005**	0.034	0.022	0.006	0.035	0.025	**0.005**	**0.027**	**0.020**	**0.005**	0.099	0.021
MH_03	0.009	0.047	0.059	0.008	0.046	0.051	**0.007**	0.039	0.047	**0.007**	**0.037**	**0.043**
MH_04	0.016	0.054	0.069	/	/	/	0.015	0.038	0.046	**0.012**	**0.036**	**0.041**
MH_05	0.013	0.093	0.050	**0.006**	0.087	0.045	**0.006**	0.083	0.045	**0.006**	**0.080**	**0.044**
V1_01	0.007	0.093	0.050	**0.006**	0.087	0.045	**0.006**	0.083	0.045	**0.006**	**0.080**	**0.044**
V1_02	**0.003**	0.049	0.053	0.006	0.037	0.038	**0.003**	0.037	0.039	**0.003**	**0.030**	**0.032**
V1_03	0.004	0.045	0.042	0.005	0.038	**0.035**	**0.003**	0.043	0.040	**0.003**	**0.037**	**0.035**
V2_01	**0.003**	**0.025**	**0.013**	0.004	0.033	0.019	**0.003**	0.028	0.017	0.004	0.034	0.016
V2_02	0.004	0.024	**0.021**	0.003	**0.024**	**0.021**	0.003	0.030	0.026	**0.002**	0.030	0.026
V2_03	**0.003**	0.042	0.036	0.005	0.043	0.039	**0.003**	0.045	0.041	**0.003**	**0.040**	0.042

Note that the bold means the best, and "/" means that cannot properly. All results include loop closure.

TABLE VII
MEAN RPE-ROTATIONAL IN UNITS OF DEGREES FOR TEN RUNS OF EACH ALGORITHM ON THE EUROC

Sequences	VINS-fusion(Stereo+IMU)			OpenVINS(Stereo+IMU)			PL-VINS			OURS		
	1 deg	1 m	1 s	1 deg	1 m	1 s	1 deg	1 m	1 s	1 deg	1 m	1 s
MH_01	0.066	0.867	0.554	0.045	**0.292**	**0.163**	0.034	0.419	0.183	**0.030**	0.387	0.182
MH_02	0.067	0.083	0.300	0.061	0.264	0.198	0.027	0.231	0.125	0.054	0.538	0.179
MH_03	0.083	0.788	0.578	0.046	**0.194**	**0.180**	**0.037**	0.244	0.210	0.055	0.343	0.275
MH_04	0.093	1.353	0.810	/	/	/	0.047	0.444	0.246	**0.034**	**0.307**	**0.167**
MH_05	0.095	0.094	0.576	0.099	0.444	0.458	**0.035**	**0.258**	0.197	**0.035**	0.265	**0.180**
V1_01	0.068	1.242	0.555	0.051	**1.263**	**0.469**	**0.016**	1.273	0.499	0.045	1.302	0.515
V1_02	0.053	0.460	0.395	0.058	0.300	0.268	0.037	0.375	0.409	**0.234**	**0.269**	**0.252**
V1_03	0.110	1.153	1.004	0.053	**0.313**	**0.262**	**0.033**	0.828	0.760	0.037	1.089	1.039
V2_01	0.188	1.288	1.060	0.036	**0.381**	**0.184**	0.039	0.585	0.323	**0.032**	0.395	0.028
V2_02	0.075	1.211	1.065	0.047	**0.432**	**0.288**	0.253	0.603	0.410	**0.018**	0.686	0.467
V2_03	0.073	2.789	2.070	0.077	**0.457**	**0.355**	0.022	0.715	0.582	**0.051**	0.774	0.634

Note that the bold means the best, and "/" means that cannot properly. All results include loop closure.

Finally, we generate a visual map of points and lines. Fig. 13 shows the visual map VINS-Fusion generated and we generated in the indoor scene (V2_02_Medium sequence), respectively. It can be seen that the point-and-line map generated by our method can better express the structured indoor environment.

B. *Evaluation in Simulation Scene*

To verify the reliability of our method in an outdoor driving environment, we chose to run VINS-Fusion, OpenVINS, PL-VINS and our method in the VIODE dataset. The VIODE [38] dataset is a data set generated by AirSim simulator in driving scenarios,

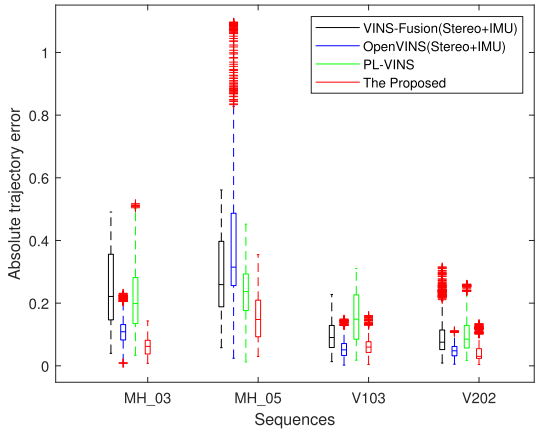


Fig. 12. Comparison of ATE box diagrams on the VIODE.

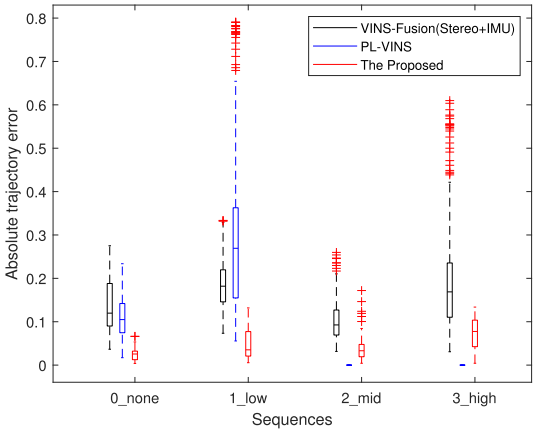


Fig. 15. Comparison of ATE box diagrams on the VIODE.

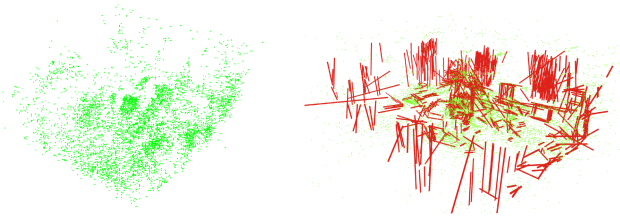


Fig. 13. Online map generation on V2_02_medium. (a) Map generated by VINS-Fusion. (b) Generated point-and-line map

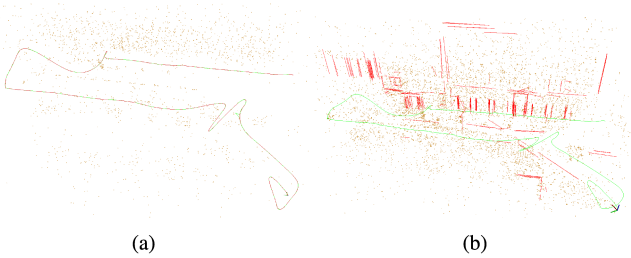


Fig. 16. Online map generation on VIODE dataset. (a) Map generated by VINS-Fusion. (b) Generated point-and-line map.

Fig. 14. The VIODE dataset Urban traffic environment.

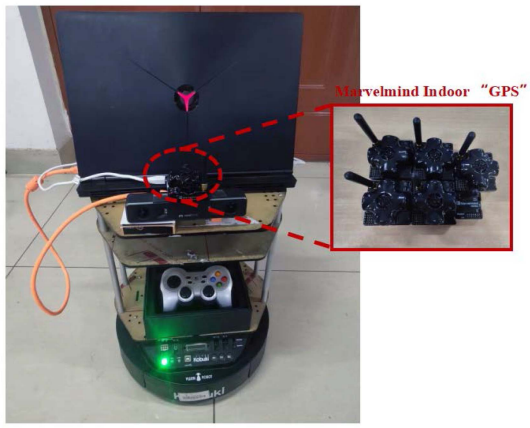


Fig. 17. The experiment platform.

which includes four dynamic levels: none, low, mid and high. This dataset includes synchronous stereo vision information and IMU information, and the running scene is shown in Fig. 14.

We run VINS-Fusion, OpenVINS, PL-INS and our algorithm in four dynamic sequences, respectively. OpenVINS can only be initialized statically and, therefore cannot run in the sequence under test. PL-VINS can not work in highly dynamic environments. We compared the running results with the real values, calculated APE and plotted the box diagram, which is shown in Fig. 15. Red boxes represent APEs of our method. It is obvious that in outdoor environment, our method can work stably.

Further, we built the points map by VINS-Fusion and the point-and-line map by our method. The results of the generated map are shown in Fig. 16. It can be seen that the point-and-line map generated by our method can express richer environmental information.

C. Evaluation in Real Environment

We verify the effectiveness of our algorithm in a real environment. As shown the experimental platform we use (See Fig. 17). The stereo camera containing the IMU is used for localization and mapping, and the Marvelmind[1] indoor "GPS" is used to obtain the Groundtruth of platform motion. In our experiment, we use Kalibr [39] to calibrate our sensors jointly.

We conduct experiments by controlling the robot to walk around a structured, low-texture indoor scene and return to the origin. We perform an online 3D reconstruction of the

[1]https://marvelmind.com/pics/marvelmindnavigation system manual.pdf.

Fig. 18. The real scene.

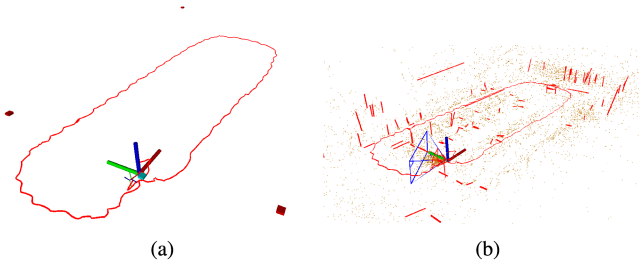


Fig. 19. Online map generation on real environment. (a) Generated trajectory. (b) Generated point-and-line map.

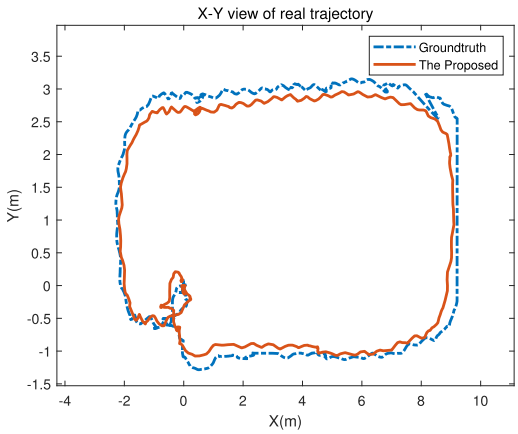


Fig. 20. Comparison of measured trajectory and Groundtruth in real environments.

low-textured interior environment (see Fig. 18). Fig. 19(a) shows the estimated trajectories of our proposed algorithm in this scenario in a real scene. The red track represents the actual track the robot walks on. Among them, the red square represents the stationary beacon as the Groundtruth measurement, and the blue square represents the moving beacon. No obvious drift was observed during the whole process. Fig. 19(b) shows the reconstructed point-and-line map. In order to quantitatively analyze the trajectory error, we draw the trajectory of the mobile beacon (Groundtruth) and the trajectory of the camera (The Proposed) respectively, and the comparison results are shown in Fig. 20. The results show that the line features can help the robot to position well in the low-textured region or even non-textured region.

VII. Conclusion

We proposed a stereo VINS based on point and line features. Using IMU information, we can more accurately determine the position of the optical flow tracking. Simultaneously, we used a hierarchical pyramid to reduce the impact of fast motion. In addition, we add real-time line features, which can help the system obtain better pose estimation in low-texture areas. The back-end redundant tightly coupled structure can continue the entire system when the stereo constraint fails. In the future, we will further integrate more sensors, such as GNSS, infrared thermal cameras and radar. At the same time, we will fuse the features of the surface, and deeply fuse the features of the point, line and surface, the endpoint of the reuse line and the endline of the surface.

References

[1] G. Huang, "Visual-inertial navigation: A. concise review," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2019, pp. 9572–9582.
[2] Z. Yu, L. Zhu, and G. Lu, "Tightly-coupled fusion of vins and motion constraint for autonomous vehicle," *IEEE Trans. Veh. Technol.*, vol. 71, no. 6, pp. 5799–5810, Jun. 2022.
[3] T. Qin, P. Li, and S. Shen, "VINS-Mono: A robust and versatile monocular visual-inertial state estimator," *IEEE Trans. Robot.*, vol. 34, no. 4, pp. 1004–1020, Aug. 2018.
[4] A. Mourikis and S. Roumeliotis, "A multi-state constraint Kalman filter for vision-aided inertial navigation," in *Proc. IEEE Int. Conf. Robot. and Automat.*, 2007, vol. 22, pp. 3565–3572.
[5] T. Qin, S. Cao, J. Pan, and S. Shen, "A general optimization-based framework for global pose estimation with multiple sensors," *CoRR*, vol. abs/1901.03642, 2019. [Online]. Available: http://arxiv.org/abs/1901.03642
[6] S. Wen, Y. Zhao, X. Liu, F. Sun, H. Lu, and Z. Wang, "Hybrid semi-dense 3D semantic-topological mapping from stereo visual-inertial odometry slam with loop closure detection," *IEEE Trans. Veh. Technol.*, vol. 69, no. 12, pp. 16057–16066, Dec. 2020.
[7] K. Sun et al., "Robust stereo visual inertial odometry for fast autonomous flight," *IEEE Robot. Automat. Lett.*, vol. 3, no. 2, pp. 965–972, Apr. 2018.
[8] S. Wen, X. Liu, H. Zhang, F. Sun, M. Sheng, and S. Fan, "Dense point cloud map construction based on stereo vins for mobile vehicles," *ISPRS J. Photogrammetry Remote Sens.*, vol. 178, pp. 328–344, 2021.
[9] R. Mur-Artal and J. D. Tardós, "ORB-SLAM2: An open-source SLAM system for monocular, stereo, and RGB-D cameras," *IEEE Trans. Robot.*, vol. 33, no. 5, pp. 1255–1262, Oct. 2017.
[10] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: An efficient alternative to SIFT or SURF," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2011, pp. 2564–2571.
[11] H. Liu, M. Chen, G. Zhang, H. Bao, and Y. Bao, "ICE-BA: Incremental, consistent and efficient bundle adjustment for visual-inertial SLAM," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 1974–1982.
[12] R. Gioi, J. Jakubowicz, J. M. Morel, and G. Randall, "LSD: A line segment detector," *Image Process. On Line*, vol. 2, no. 4, pp. 35–55, 2012.
[13] L. Zhang and R. Koch, "An efficient and robust line segment matching approach based on LBD descriptor and pairwise geometric consistency," *J. Vis. Commun. Image Representation*, vol. 24, pp. 794–805, 2013.
[14] Q. Fu et al., "PL-VINS: Real-Time Monocular Visual-Inertial SLAM with Point and Line Features," 2020, *arXiv:2009.07462*.
[15] J. Engel, T. Schoeps, and D. Cremers, "LSD-SLAM: Large-scale direct monocular SLAM," in *Proc. Eur. Conf. Comput. Vis.*, 2014,pp. 1–16.
[16] J. Engel, V. Koltun, and D. Cremers, "Direct sparse odometry," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 3, pp. 611–625, Mar. 2018.
[17] C. Campos, R. Elvira, J. J. G. Rodríguez, J. M. M. Montiel, and J. D. Tardós, "ORB-SLAM3: An Accurate Open-Source Library for Visual, Visual–Inertial, and Multimap SLAM," *IEEE Trans. Robot.*, vol. 37, no. 6, pp. 1874–1890, Dec. 2021.

[18] P. Geneva, K. Eckenhoff, W. Lee, Y. Yang, and G. Huang, "Openvins: A research platform for visual-inertial estimation," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2020, pp. 4666–4672.
[19] R. Gomez-Ojeda, F.-A. Moreno, D. Zuñiga-Noël, D. Scaramuzza, and J. Gonzalez-Jimenez, "PL-SLAM: A. stereo SLAM system through the combination of points and line segments," *IEEE Trans. Robot.*, vol. 35, no. 3, pp. 734–746, Jun. 2019.
[20] J. Shi and Tomasi, "Good features to track," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 1994, pp. 593–600.
[21] H. Xie, W. Chen, J. Wang, and H. Wang, "Hierarchical quadtree feature optical flow tracking based sparse pose-graph visual-inertial SLAM," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2020, pp. 58–64.
[22] R. Gomez-Ojeda, J. Briales, and J. Gonzalez-Jimenez, "PL-SVO: Semi-direct monocular visual odometry by combining points and line segments," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2016, pp. 4211–4216.
[23] C. Zhang, "PL-GM:RGB-D SLAM with a novel 2D and 3D geometric constraint model of point and line features," *IEEE Access*, vol. 9, pp. 9958–9971, 2021.
[24] Y. He, J. Zhao, Y. Guo, W. He, and K. Yuan, "Pl-vio: Tightly-coupled monocular visual–inertial odometry using point and line features," *Sensors*, vol. 18, no. 4, 2018, Art. no. 1159.
[25] H. Lim, J. Jeon, and H. Myung, "UV-SLAM: Unconstrained line-based slam using vanishing points for structural mapping," *IEEE Robot. Automat. Lett.*, vol. 7, no. 2, pp. 1518–1525, Apr. 2022.
[26] H. Wei, F. Tang, C. Zhang, and Y. Wu, "Highly efficient line segment tracking with an IMU-KLT prediction and a convex geometric distance minimization," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2021, pp. 3999–4005.
[27] X. Zuo, X. Xie, Y. Liu, and G. Huang, "Robust visual SLAM with point and line features," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2017, pp. 1775–1782.
[28] J. Lee and S.-Y. Park, "PLF-VINS: Real-time monocular visual-inertial SLAM with point-line fusion and parallel-line fusion," *IEEE Robot. Automat. Lett.*, vol. 6, no. 4, pp. 7033–7040, Oct. 2021.
[29] W. Baird, "An introduction to inertial navigation," *Amer. J. Phys.*, vol. 77, pp. 844–847, 2009.
[30] C. Forster, L. Carlone, F. Dellaert, and D. Scaramuzza, "IMU preintegration on manifold for efficient visual-inertial maximum-a-posteriori estimation," in *Proc. 11th Conf. Robot. Sci. Syst.*, Jul. 2015, pp. 1–10.
[31] S. Baker, R. Gross, and I. Matthews, "Lucas-kanade 20 years on: A. unifying framework," *Int. J. Comput. Vis.*, vol. 56, pp. 221–255, 2004.
[32] J. Canny, "A computational approach to edge detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-8, no. 6, pp. 679–698, Nov. 1986.
[33] D. Ballard, "Generalizing the hough transform to detect arbitrary shapes," *Pattern Recognit.*, vol. 13, no. 2, pp. 111–122, 1981. [Online]. Available: https://www.sciencedirect.com/science/article/pii/0031320381900091
[34] Huber and J. Peter, "Robust estimation of a location parameter," *Ann. Math. Statist.*, vol. 35, no. 1, pp. 73–101, 1964.
[35] G. Zhang, J. H. Lee, J. Lim, and I. H. Suh, "Building a 3-D line-based map using stereo SLAM," *IEEE Trans. Robot.*, vol. 31, no. 6, pp. 1364–1377, Dec. 2015.
[36] M. Burri et al., "The EuRoC micro aerial vehicle datasets," *Int. J. Robot. Res.*, vol. 35, no. 10, pp. 1157–1163, 2016.
[37] Jürgen Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A benchmark for the evaluation of RGB-D SLAM systems," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2012, pp. 573–580.
[38] K. Minoda, F. Schilling, V. Wüest, D. Floreano, and T. Yairi, "VIODE: A simulated dataset to address the challenges of visual-inertial odometry in dynamic environments," *IEEE Robot. Automat. Lett.*, vol. 6, no. 2, pp. 1343–1350, Apr. 2021.
[39] J. Rehder, J. Nikolic, T. Schneider, T. Hinzmann, and R. Siegwart, "Extending kalibr: Calibrating the extrinsics of multiple IMUs and of individual axes," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2016, pp. 4304–4311.

Xin Liu was received the B.Sc. degree in automation from the Yanshan University of Science and Technology, Qinhuangdao, China. He is currently working toward the Ph.D. degree with the Department of Electric Engineering, Yanshan University, Qinhuangdao. His research interests include SLAM and multi-sensor information fusion.

Shuhuan Wen (Senior Member, IEEE) received the Ph.D. degree in control theory and control engineering from the Yanshan University, Qinhuangdao, China, in 2005. She is currently a Professor of automatic control with the Department of Electric Engineering, Yanshan University. She has coauthored one book, about 40 papers. Her research interests include SLAM, computer vision, robotics, 3-D object recognization and reconstruction.

Dr. Wen was a Visiting Scholar of the Ottawa University, Carleton University and Simon Fraser University in Canada from 2011 to 2013. She is also a visiting professor at University of Alberta in Canada from 2021 to 2022. She is an associate editor of IROS from 2021 to 2022.

Hong Zhang (Fellow, IEEE) received the B.Sc. degree in electrical engineering from Northeastern University, Boston, MA, USA, in 1982, and the Ph.D. degree in electrical engineering from Purdue University, West Lafayette, IN, USA, in 1986. He conducted Postdoctoral research with the University of Pennsylvania, Philadelphia, PA 19104, USA, from 1986 to 1987, before joining the Department of Computing Science, University of Alberta, Edmonton, AB, Canada. He is currently the NSERC Industrial Research Chair. His research interests include robotics computer vision and image processing.