LJMU Research Online

Jiang, H, Cang, N, Lin, Y, Guo, D and Zhang, W

 LR-SLAM: Visual Inertial SLAM System with Redundant Line Feature Elimination

http://researchonline.ljmu.ac.uk/id/eprint/26003/

Article

For more information please contact researchonline@ljmu.ac.uk

LR-SLAM: Visual Inertial SLAM System with Redundant Line Feature Elimination

Hao Jiang[1,2] · Naimeng Cang[1,2] · Yuan Lin[1,2] · Dongsheng Guo[1,2] · Weidong Zhang[1,2]

Abstract

The present study focuses on the simultaneous localization and mapping (SLAM) system based on point and line features. Aiming to address the prevalent issue of repeated detection during line feature extraction in low-texture environments, a novel method for merging redundant line features is proposed. This method effectively mitigates the problem of increased initial pose estimation error that arises when the same line is erroneously detected as multiple lines in adjacent frames. Furthermore, recognizing the potential for the introduction of line features to prolong the marginalization process of the information matrix, optimization strategies are employed to accelerate this process. Additionally, to tackle the issue of insufficient point feature accuracy, subpixel technology is introduced to enhance the precision of point features, thereby further reducing errors. Experimental results on the European Robotics Challenge (EUROC) public dataset demonstrate that the proposed LR-SLAM system exhibits significant advantages over mainstream SLAM systems such as ORB-SLAM3, VINS-Mono, and PL-VIO in terms of accuracy, efficiency, and robustness.

1 Introduction

In recent years, the technology of SLAM (Simultaneous Localization and Mapping) [1] has garnered substantial attention owing to its extensive applicability, notably in the domains of autonomous vehicles [2, 3], robotics [4, 5], and Unmanned Aerial Vehicles [6]. Among the pivotal sensors employed in this technology, cameras are fundamental, primarily categorized into three distinct types: stereo cameras, RGB-D cameras [7], and monocular cameras. In terms of their inherent capabilities, stereo cameras provide a direct methodology for pose estimation by leveraging a defined baseline [8, 9] to ascertain the spatial coordinates of each pixel. Nevertheless, they are hindered by the intricacies involved in their configuration and calibration procedures, as well as the constraints on depth range and accuracy imposed by the stereo baseline and resolution limitations [10]. Conversely, RGB-D cameras possess the capability to directly ascertain the distance between objects and the camera, thereby facilitating direct pose estimation. However, their utilization is encumbered by a narrow measurement range, elevated noise levels, and the susceptibility to frequent disruptions stemming from sunlight [11].

The field of monocular vision-based SLAM boasts numerous seminal algorithms, including ORB-SLAM2 [12] (Oriented FAST and Rotated BRIEF for SLAM2), VINS-Mono [13] (Visual-Inertial Navigation System Mono), the monocular-oriented PL-SLAM [14] (Real-Time Monocular Visual SLAM Leveraging Points and Lines), PL-VIO [15] (Tightly-Coupled Monocular Visual-Inertial Odome-

✉ Dongsheng Guo
    gdongsh2022@hainanu.edu.cn

    Hao Jiang
    kl1094535240@163.com

    Naimeng Cang
    nmcang@hainanu.edu.cn

    Yuan Lin
    yuanlin@hainanu.edu.cn

    Weidong Zhang
    wdzhang@sjtu.edu.cn

[1] School of Information and Communication Engineering, Hainan University, Haikou 570228, China

[2] Engineering Research Center of Marine Intelligent Systems, Ministry of Education, Haikou 570228, China

try Incorporating Point and Line Features), and PL-VINS [16] (Point and Line Visual-Inertial Navigation System). These methodologies are adaptable to diverse environments and varying illumination condition, albeit they are prone to limitations in feature extraction qualities [17]. Notably, ORB-SLAM2, by relying solely on a single image feature as a constraint for pose estimation, may exhibit diminished robustness in certain scenarios.

To mitigate the disruptions stemming from the exclusive reliance on a single feature, inertial measurement units (IMU) have been seamlessly integrated into the original monocular systems, notably VINS-Mono and the subsequent iteration of ORB-SLAM2. By harnessing the complementary strengths of two distinct sensors, these algorithms demonstrate enhanced capabilities in handling environments characterized by weak or absent textures, as compared to ORB-SLAM2. However, it is noteworthy that VINS-Mono, which employs optical flow for feature point tracking, remains vulnerable to fluctuations in lighting conditions. Similarly, ORB-SLAM3 [18] inherits this limitation from ORB-SLAM2. In an effort to address these challenges, researchers have augmented existing algorithms with line features, thereby providing additional constraints for pose estimation. This has led to the emergence of systems such as PL-VIO, PL-SLAM, and PL-VINS. Yet, the incorporation of line features into SLAM systems is not without its complexities. One prominent issue is the occurrence of erroneous line feature matching due to repeated detections. Classical line feature extraction methods, like LSD (Line Segment Detector), are susceptible to this problem. In some instances, LSD may erroneously segment a single, continuous straight line into multiple shorter segments within the image, leading the system to recognize them as distinct line features. These mistakenly distinguished line features subsequently participate in feature matching across consecutive frames, potentially introducing significant errors into the pose estimation process. This issue arises from the fact that line features are inherently derived from point features, and during the extraction phase, they can be distorted by noise and other confounding factors.

Apart from the aforementioned issues with line features, the integration of line feature information in SLAM systems necessitates the use of an information matrix for pose estimation that becomes more computationally intensive, requiring additional optimization measures. Additionally, the corner points extracted by these systems are inherently constrained to pixel-level precision, whereas the actual positions of the corners often do not coincide with integer pixel positions. To address these concerns, this paper introduces the LR-SLAM algorithm, which builds upon and improves upon the PL-VIO framework. The key innovative contributions of the LR-SLAM algorithm are outlined as follows:

(1) By employing a sub-pixel corner point detection tech-

nique and applying sub-pixel edge constraints to corner points, we effectively enhance the matching precision of feature points. (2) This work represents the initial endeavor to utilize a combined approach of 2D and 3D distance assessments for the elimination of redundant line features. Initially, we discern the unique ID number assigned to each line feature. Subsequently, we evaluate the 3D distance along the structured line feature, along with the angle and distance in 2D. From these three perspectives, we assess whether the line features fulfill the criteria for merging, and subsequently merge those that do, thereby achieving the removal of redundant line features. (3) The proposed method is primarily optimized for the marginalization of the information matrix. In cases where the information to be marginalized encompasses camera pose information, the information matrix undergoes a two-step marginalization process followed by integration. Conversely, when this is not the case, the entire matrix undergoes marginalization.

Figure 1 shows an example of LR-SLAM output for sequence MH01 in the EUROC dataset, where the sub-pixel point feature (one frame) of sequence MH01 in the EUROC dataset is shown on the left in Fig. 1(a), and the line feature (one frame) of the merged sequence MH01 in the EUROC dataset is shown on the right in the Fig. 1(b). In Fig. 1(a) shown are the feature points extracted from the image, which are represented using red and blue color respectively, the redder the feature point the more frames are being tracked, and the bluer the feature point the lesser number of frames are being tracked. The rest of the paper is organized as follows: related work on SLAM is presented in Section 2; Section 3 details the approach proposed in this paper; Section 4 gives a comparison between several common SLAM systems and the SLAM system proposed in this paper on the EUROC dataset experiments; Section 5, draws conclusions.

2 Related Works

With the development of SLAM technology, many visual SLAM methods have been investigated and some representative methods have emerged, such as the feature point-based ORB-SLAM3 [16], the PL-VIO [14] based on point-line features, and VINS-Mono [13]. A robust SLAM system is essential for the further development of the field.

2.1 Feature Point Extraction

In feature-based SLAM, feature points are crucial. ORB-SLAM3 extracts points from an image pyramid using FAST and describes them with BRIEF [19], ensuring robustness and accuracy. PL-SLAM follows a similar approach. Meanwhile, PL-VIO detects parallel lines to extract geometrically constrained feature points, improving matching robustness
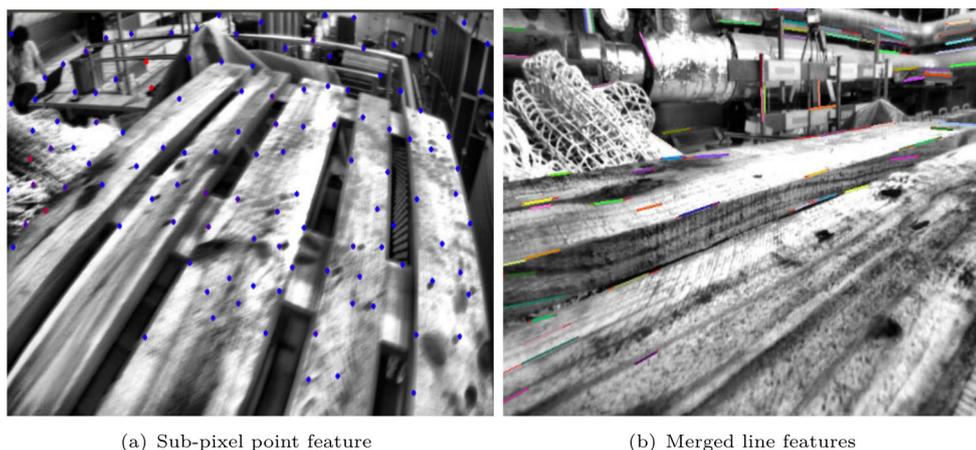
(a) Sub-pixel point feature   (b) Merged line features

Fig. 1 The extracted point and line features of the LR-SLAM system in one frame image of the MH01 sequence are respectively displayed

and efficiency. VINS-Mono uses FAST for ORB feature extraction, but these methods can struggle with accuracy and motion blur. This thesis adopts a sub-pixel method to improve precision and resilience to motion blur.

2.2 Line Feature Extraction

Compared to point features, line features in SLAM systems [20] exhibit three primary advantages elaborated: they provide richer geometric constraint information, including angles and lengths between segments, enhancing camera motion estimation accuracy and map quality; they excel in low-texture regions due to their stability against illumination and occlusion changes; and they offer higher measurement precision, determined by more pixels, leading to a more robust and accurate optimization process. In PL-VIO and PL-SLAM, LSD [21] is the primary line feature detector, while PL-VINS employs an improved LSD with optimized hidden parameters and length suppression. However, these methods may suffer from duplicate line detections, increasing pose estimation errors. To mitigate this, the proposed line feature merging method aims to reduce such duplications for more accurate pose estimation.

2.3 Marginalization of Information Matrix

The simplest solution for the relative camera pose is obtained by calculating it from two frames before and after the image. Although this method is fast but low in accuracy and can only be used in a short period of time, it is highly accurate but inefficient if a global optimization method (such as bundle adjustment) is used [22]. Therefore, researchers have proposed the sliding window approach [23], which performs optimization operations on a fixed number of frames at a time, thus ensuring both accuracy and efficiency. Since it is a sliding window, new image frames are bound to come in as well

as old ones leave during the sliding process. Marginalization is a technique aimed at effectively utilizing the discarded image frames. It involves retaining valuable information, such as prior knowledge and IMU data, from those frames that are no longer needed, ensuring that no useful data goes to waste. In PL-VIO system, in order to speed up the marginalization process, the marginalization of information matrix is divided into two steps. Firstly, marginalization of the part except camera pose, then marginalization of the camera pose, and finally merging the two parts.

3 Proposed Method

This section consists of four parts. First, the thematic framework of the LR-SLAM algorithm is briefly introduced. Next, the sub-pixel method for optimizing feature points will be introduced. Then, redundant line removal and its application in line feature detection process will be discussed. Finally, the optimization method of marginalized information matrix will be presented.

3.1 System Framework

The system framework demonstrated in Fig. 2 represents the main framework of the LR-SLAM algorithm, which is based on the PL-VIO algorithm with additional improvements highlighted in yellow boxes. The system framework is divided into two main components: the front-end and the back-end. The front-end utilizes extracted feature lines and IMU data for preliminary position estimation. On the other hand, the back-end is responsible for refining the position accuracy and optimizing it. This algorithm achieves localization based on the following equations.

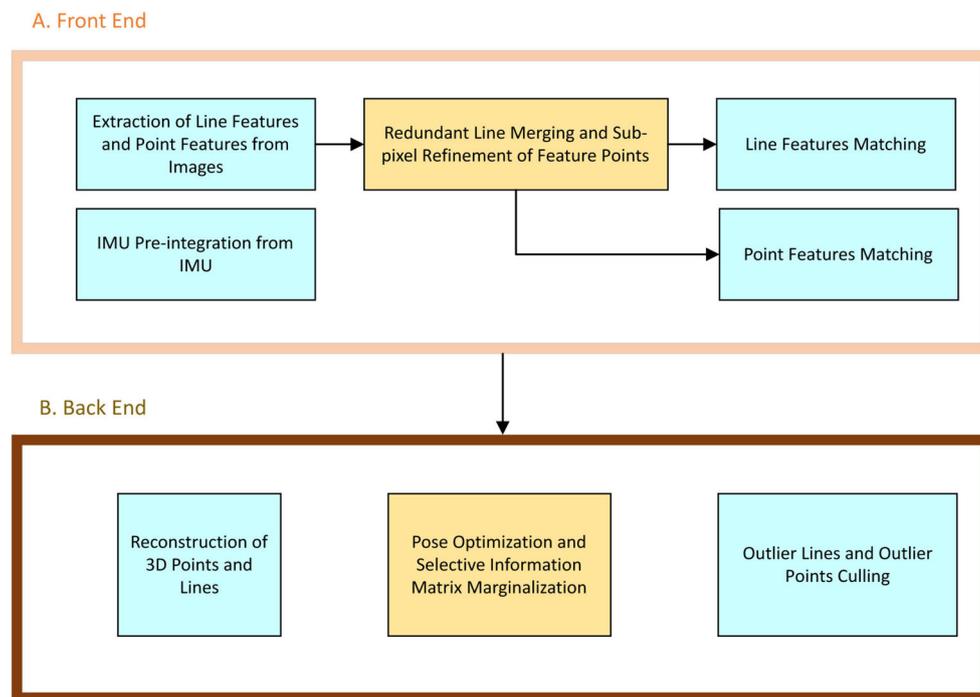$$\begin{cases} A_k = f(A_{k-1}, U_k) + W_l, \\ C_{k,j} = h(B_j, A_k) + V_{k,j}, \end{cases} \tag{1}$$

A. Front End

```
┌─────────────────────────────────────────────────────────────────────────────┐
│  ┌──────────────────┐    ┌──────────────────┐         ┌──────────────────┐   │
│  │ Extraction of Line│   │ Redundant Line   │         │ Line Features    │   │
│  │ Features and Point│──▶│ Merging and Sub- │────────▶│ Matching         │   │
│  │ Features from     │   │ pixel Refinement │         │                  │   │
│  │ Images            │   │ of Feature Points│         └──────────────────┘   │
│  └──────────────────┘    └──────────────────┘                               │
│  ┌──────────────────┐            │                    ┌──────────────────┐   │
│  │ IMU Pre-integration│          └───────────────────▶│ Point Features   │   │
│  │ from IMU          │                                │ Matching         │   │
│  └──────────────────┘                                 └──────────────────┘   │
└─────────────────────────────────────────────────────────────────────────────┘
```

B. Back End

```
┌─────────────────────────────────────────────────────────────────────────────┐
│  ┌──────────────────┐    ┌──────────────────┐         ┌──────────────────┐   │
│  │ Reconstruction of│    │ Pose Optimization│         │ Outlier Lines and│   │
│  │ 3D Points and    │    │ and Selective    │         │ Outlier Points   │   │
│  │ Lines            │    │ Information      │         │ Culling          │   │
│  │                  │    │ Matrix           │         │                  │   │
│  │                  │    │ Marginalization  │         │                  │   │
│  └──────────────────┘    └──────────────────┘         └──────────────────┘   │
└─────────────────────────────────────────────────────────────────────────────┘
```

Fig. 2 Systematic framework for the LR-SLAM algorithm

where, A represents the position at a certain time, B represents an observed point, C represents the pixel mapping of the observed point in the image, and U represents known data. f and h represent the functional relationship between them, while W and V denote noise.

3.2 Sub-pixel Optimization of Feature Points

Although the PL-VIO system has many features, there are still many problems in system implementation. For example, the pixel coordinates of feature points are integers and have a certain deviation from the actual image, which can affect the accuracy of feature point matching. Therefore, in this paper, LR-SLAM is proposed to improve the accuracy of feature points using the sub-pixel method. By utilizing the sub-pixel method, the feature points' coordinates can be refined from integers to decimals, thereby improving the matching accuracy between features.

First get enough corner point and initialization results. If the number of feature corner point is less than the set parameter value of 150, additional feature corner point are needed. If the number of feature corner point is more than 150, the 150 corner point with the best quality among them are selected. Then the gradient information of the image gray value is used to interpolate the feature corner point positions to get the position coordinates at the sub-pixel level, and a Gaussian Newton iteration method is used to optimize the coordinate

values. The pixel coordinate values of the new feature corner point are finally obtained.

Of course, since the image is bounded, the feature corner point are optimized in the sub-pixel to ensure that the optimized corner point coordinate values cannot exceed the image boundaries. The method used here is to compare the sub-pixel coordinates with the pixel coordinates of the image boundary, and thus to constrain the sub-pixel corner point to cross the boundary. The specific steps are as follows: 1.Iterate through all sub-pixelated corner point features; 2.Let the x and y values of the corner point feature pixel coordinates be compared with the 1-value point (1-value point refers to the point where x and y are both 1) and the sub-maximum point (x is the column value of the image minus 1, y is the row value of the image minus 1) of the image in the same coordinate system, if the x or y in the feature corner point pixel coordinates is smaller than the x or y of the 1-value point, then the x or y in the pixel coordinates is the x or y of the sub-maximum point. If the x or y in the pixel coordinates of the feature corner point is greater than the x or y of the sub-maximum point, the x or y in the pixel coordinates is the x or y of the sub-maximum point.

3.3 Redundant Line Deletion Method

In order to reduce the increase of pose estimation error caused by continuously detecting line features in adjacent frames,

this paper proposes a redundant line removal method (as shown in Fig. 3). This method is divided into three steps in total, which will be elaborated in the following three sub-sections. This is also the focus and difficulty of this study. The final output of line features is shown in Fig. 4(a) (while Fig. 4(b) shows the initial line features).

3.3.1 Line Feature ID Determination

The LSD algorithm is mainly used in this paper to implement line segment detection and extraction. Line segments with a certain length and direction are detected in the image, and line segment descriptors are used to represent the features of these line segments. A unique ID is also assigned to each line feature for subsequent tracking and matching. Based on the property of the line feature IDs, screening can be performed by the ID number of the line feature, since only old IDs may be duplicate lines of new IDs. This screening is also called coarse screening. After the coarse screening, the line feature that meet the requirements will be screened in the following two steps, which are also called fine screening.

3.3.2 Distance Judgment of Line Features in Three-dimensional Space

The distance judgment of line feature in 3D space is the first step of fine screening; first, the line feature under the camera coordinate system are structured; then, the distance judgment

Fig. 3 Flow chart of redundant line deletion method

of the structured line feature is performed; finally, the two line feature whose distance is smaller than the set threshold are subjected to the last step of fine screening. The specific steps are detailed as follows:

(1) Line feature structure

Based on the information contained in the line feature, e.g., the length and angle of the line feature, the point coordinate values of the line feature in camera coordinates are calculated (two-dimensionalized):

$$\begin{cases} x_l = d_l \cos \theta_l, \\ y_l = d_l \cos \theta_l, \end{cases} \tag{2}$$

where (x_l, y_l) is the point coordinate, θ_l is the angle of the line feature, and d_l is the length of the line feature. When the point coordinates are known, the z-axis values of the line features in the camera coordinate system are added to make the point coordinates up-dimensional. Since the line feature has a front end and a back end, there are two 3D point coordinates after the dimensioning, $c_l^1(x_l^1, y_l^1, z_l^1)$ and $c_l^2(x_l^2, y_l^2, z_l^2)$. Then the selection matrix is converted to a three-dimensional Manhattan coordinate system with two coordinates $c_{M_l}^1(x_{M_l}^1, y_{M_l}^1, z_{M_l}^1)$ and $c_{M_l}^2(x_{M_l}^2, y_{M_l}^2, z_{M_l}^2)$, respectively.

(2) Line Feature Distance Determination on a Three-dimensional Plane

The endpoint coordinates of the two line features in the Manhattan coordinate system are obtained in the above manner. By subtracting the corresponding endpoint coordinates, the line vectors l_1 and l_2 representing the two lines are obtained as follows:

$$\begin{cases} (x_{l_1}^2 - x_{l_1}^1, y_{l_1}^2 - y_{l_1}^1, z_{l_1}^2 - z_{l_1}^1), \\ (x_{l_2}^2 - x_{l_2}^1, y_{l_2}^2 - y_{l_2}^1, z_{l_2}^2 - z_{l_2}^1), \end{cases} \tag{3}$$

$$\begin{cases} dis_1 = ||p \times l_1||_2 / ||l_1||_2, \\ dis_2 = ||p \times l_2||_2 / ||l_2||_2 \end{cases} \tag{4}$$

$$dis = (dis_1 + dis_2)/2 \tag{5}$$

The auxiliary vector p_1 is then constructed in the above way based on the coordinates of the same end of the two line feature (either end is chosen, in this paper the coordinates of the front end are chosen). The distance values of the two lines are calculated from Eq. 4 as dis_1 and dis_2, and then the final distance value between the two lines is derived from Eq. 5 as dis. According to the obtained distance value, dis is compared with the threshold value 0.1, if it is lower, the next step of fine screening is performed.

(3) Line feature distance determination on a two-dimensional plane

Two 3D line features are projected into 2D space to generate two 2D lines. The endpoint coordinates of the two-dimensional lines are also both two-dimensional. At this
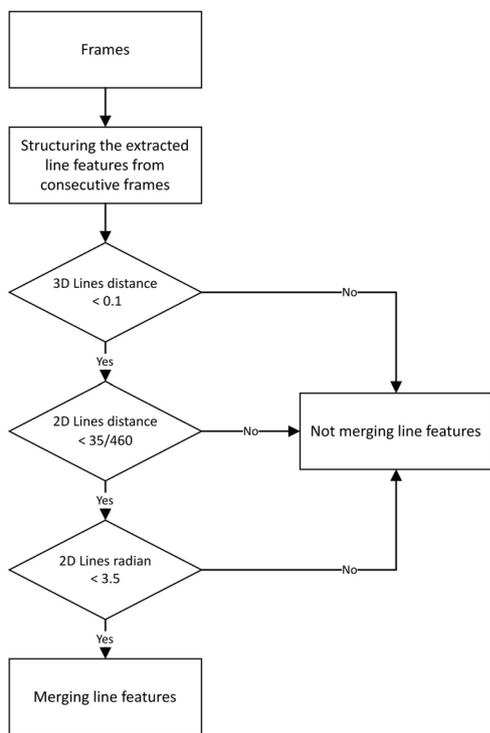
Fig. 4 (a) It shows the initially extracted linear features (in red). (b)It shows the merged linear features (in white). Images are sourced from the KITTI dataset(100 features)

(a)

(b)

point, the endpoint coordinates of the two two-dimensional lines are expanded (in three dimensions) to construct their flush coordinates, $c_{l1}^1 (x_{1^1}^1, y_{1^1}^1, 1), c_{l1}^2 (x_{l1}^2, y_{l1}^2, 1)$ and $c_{l2}^1 (x_{1^2}^1, y_{1^2}^1, 1), c_{l2}^2 (x_{l2}^2, y_{l2}^2, 1)$, respectively. Select the two chisquare coordinates corresponding to one of the lines (the first line is chosen in this paper) to determine a linear line vector p_2.

$$\begin{cases} dis_3 = |p_2 \cdot c_{l2}^1|/||p_2||_2, \\ dis_4 = |p_2 \cdot c_{l2}^1|/||p_2||_2, \end{cases} \tag{6}$$

The flush endpoint coordinates of the other line to the line vector distance l_p is given by the Eq. 6 for dis_3 and dis_4. By determining whether one of dis_3 and dis_4 is smaller than the threshold 35/460, if it is satisfied, the angle comparison of the two lines l_1 and l_2 is performed, and their direction vectors are obtained by means of the Eq. 3.

$$\arccos \frac{l_1 \cdot l_2}{\sqrt{l_1 \cdot l_1} \cdot \sqrt{l_2 \cdot l_2}} \tag{7}$$

The number of angles between the two lines is calculated according to the Eq. 7. The obtained angle number is converted into radian number, and if the radian number is less than 3.5, the line features are merged to achieve the purpose of redundant line deletion.

3.4 Marginalization of the Information Matrix

In SLAM systems, when a new observation is added, a new node and some new edges are created, and an information matrix can describe the relationship between these nodes and edges. In PL-VIO, the information matrix is a matrix used to describe the relationship between several variables such as camera pose, IMU measurement, and feature point location.

However, in practical applications, the information matrix can significantly expand in size, resulting in substantial computational overhead. Therefore, some strategies need to be adopted to simplify the problem. One of the solutions is to use Schur complement to marginalize the information matrix, thus reducing the solution volume and optimization cost. Marginalization of the information matrix transforms the part of the information matrix related to the unknown variables into a new information matrix by matrix elimination based on the known variables, simplifying the problem.

This paper proposes an optional information matrix marginalization method based on the PL-VIO marginalization method. The optional information matrix marginalization method adds selectivity to the traditional information matrix marginalization. The traditional information matrix marginalization involves marginalizing the entire matrix, while the optional method allows for marginalization based on different situations. Therefore, when the marginalized information contains complex camera pose information, the marginalization is performed in two steps: first, the infor-

mation matrix of non-camera poses is marginalized; then, the information of camera poses is marginalized. Finally, the marginalized information from the first two steps is merged. If the marginalized information does not contain camera pose information, the entire information matrix is marginalized. This method helps improve the efficiency of information matrix marginalization.

The marginalization of the non-camera pose information matrix mentioned above is performed to marginalize the pose and motion information of the IMU. This non-camera state information includes the angle, angular velocity, and linear acceleration of the IMU. First, using the angular velocity and linear acceleration measurements of the IMU, the changes of the IMU states (e.g., attitude, velocity, and displacement) over a time window are computed using a pre-integration method; then, based on some criteria (e.g., relevance, frequency of use, etc.), a decision is made as to which IMU states or observations will be marginalized; and, finally, the variables that need to be marginalized are removed from the information matrix and from the right-hand side observation vectors.

4 Experiment

In this section, the method proposed in this paper is compared with open-source algorithms such as VINS-Mono, ORB-SLAM3's monocular and PL-VIO, tested on the EUROC dataset [24] and the KITTI dataset [25], and evaluated with the EVO Evaluation Tool on the trajectory maps of the above SLAM systems.

4.1 Experimental Platform

The experiments were conducted on Ubuntu 18.04. The main hardware used in this article includes Intel(R) Core(TM) i7-9750H CPU @ 2.60 GHz, 3.0 GHz 16 GB and NVIDA GeForce GTX 2060.

4.2 Setting Parameters

The optimized corner points in LR-SLAM are 150, the image resolution is, the distance scale factor of the line features in 3D space is 0.1, the distance scale factor of the line features in 2D plane is 35/460, and the angle degree between the line features.

4.3 Evaluation Metrics

In this paper, the evaluation metric ATE is used to evaluate the experimental results of each SLAM system, which measures the accuracy of the algorithm and the global consistency of the trajectory [26]. The subset of ATE metrics consists of

Table 1 Metric absolute trajectory error (ATE/m) results for VINS-Mono and LR-SLAM on the EUROC dataset (MH series)

Data	VINS-Mono			LR-SLAM		
	Mean	Std	Rmse	Mean	Std	Rmse
MH01	0.188	0.110	0.218	0.111	0.055	0.124
MH03	0.341	0.255	0.426	0.217	0.094	0.237
MH04	0.334	0.230	0.406	0.302	0.088	0.314
MH05	0.322	0.206	0.383	0.266	0.071	0.276

certain metrics, among which, the dominant metrics are root mean square error (RMSE), mean absolute error (MEAN) [27]. The following illustrates the detailed definitions of all three main indicators in ATE:

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(J(m_i) - n_i)^2}, \tag{8}$$

$$MEAN = \frac{1}{N}\sum_{i=1}^{N}|J(m_i) - n_i|, \tag{9}$$

$$STD = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(J(m_i) - Mean)^2}, \tag{10}$$

where m and n are respectively the keyframe trajectory generated by the system and the ground truth associated with them, N is the number of keyframes, i is the keyframe sequence number, and $J(\cdot)$ is a trajectory alignment function based on rotation and scale consistency between m_i and n_i.

4.4 EUROC and KITTI Dataset

In order to better validate the performance of the developed LR-SLAM method and compare it with other excellent SLAM systems, several excellent sequences provided by the widely used EUROC public dataset were selected for testing. In this paper, four MH sequences from the EORUC dataset (one simple sequence MH01, one medium sequence MH03, and two difficult sequences MH04 and MH05) as well as a partial dataset from KITTI were chosen. These data can be

Table 2 Metric absolute trajectory error (ATE/m) results for ORB-SLAM3 monocular and LR-SLAM on the EUROC dataset (MH sequence)

Data	ORB-SLAM3			LR-SLAM		
	Mean	Std	Rmse	Mean	Std	Rmse
MH01	1.977	1.123	2.274	0.111	0.055	0.124
MH03	2.878	1.203	3.120	0.217	0.094	0.237
MH04	6.084	2.333	6.516	0.302	0.088	0.314
MH05	3.398	2.060	4.005	0.266	0.071	0.276

**Table 3** Metric absolute trajectory error (ATE/m) results for PL-VIO and LR-SLAM on the EUROC dataset (MH series)

Data	PL-VIO			LR-SLAM		
	Mean	Std	Rmse	Mean	Std	Rmse
MH01	0.153	0.084	0.175	0.111	0.055	0.124
MH03	0.238	0.110	0.262	0.217	0.094	0.237
MH04	0.336	0.104	0.352	0.302	0.088	0.314
MH05	0.323	0.210	0.385	0.266	0.071	0.276

Table 4 Metric absolute trajectory error (ATE/m) results for PL-VIO and LR-SLAM on the EUROC dataset (MH series)

Data	VINS-Mono			ORB-SLAM3			PL-VIO		
	Mean	Std	Rmse	Mean	Std	Rmse	Mean	Std	Rmse
MH01	40.9	50.0	42.9	94.4	95.1	94.5	27.2	34.5	28.6
MH03	36.0	63.1	44.4	92.4	92.1	92.4	8.4	15.3	9.9
MH04	9.8	61.7	22.4	95.0	96.2	95.1	10.1	15.4	10.5
MH05	17.3	65.7	28.0	92.1	96.5	93.1	17.3	66.3	28.5

Table 5 Metric absolute trajectory error (ATE/m) results for PL-VIO and LR-SLAM on the KITTI dataset (part series)

Data	VINS-Mono			ORB-SLAM3			PL-VIO		
	Mean	Std	Rmse	Mean	Std	Rmse	Mean	Std	Rmse
KITTI00	9.6	11.0	8.9	86.5	82.1	83.3	7.4	6.9	8.4
KITTI02	2.3	4.3	/	87.4	72.1	76.7	8.4	12.9	6.5
KITTI04	14.8	22.7	19.6	82.3	89.7	84.3	8.1	7.8	9.1
KITTI06	13.1	5.7	6.8	89.7	91.5	84.3	7.7	6.9	9.2

Fig. 5 (a) Compare the trajectory predictions of VINS-Mono (blue), ORB-SLAM3 monocular (green), PL-VIO (red), and LR-SLAM (purple) for sequence MH03 with the ground truth trajectory (brown). (b), (c) and (d) respectively show the comparisons between VINS-Mono, ORB-SLAM3, and PL-VIO (green) with LR-SLAM (blue) and the ground truth (brown)
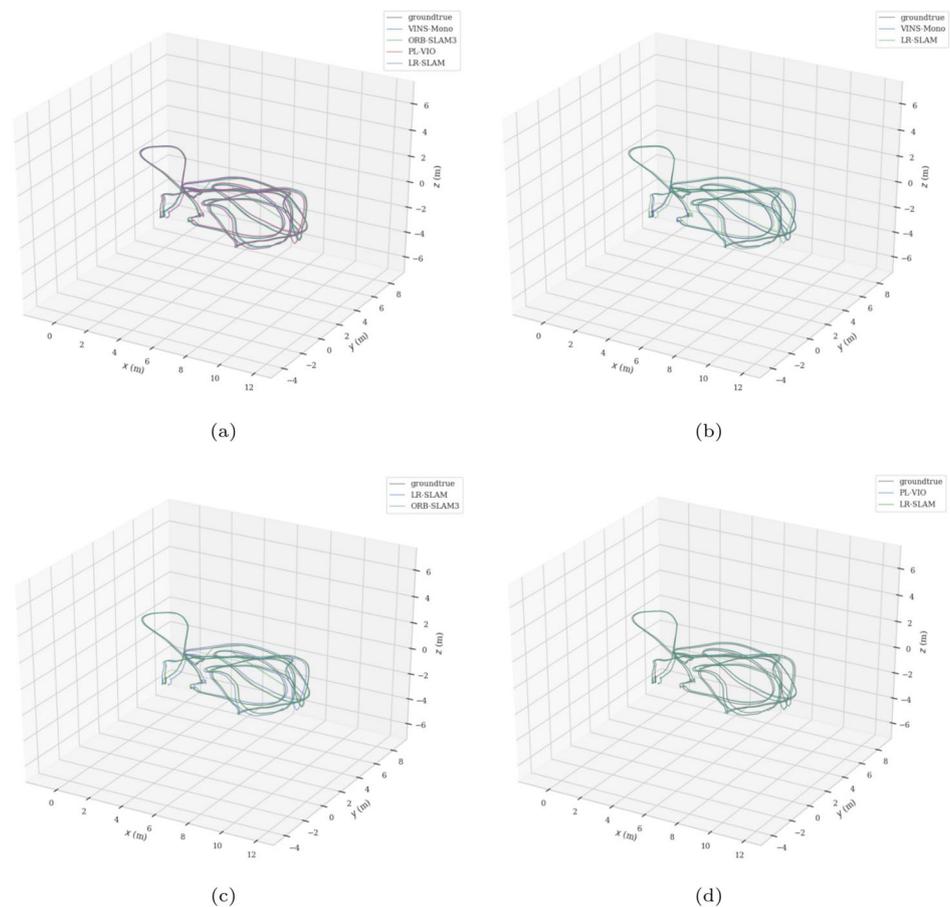
used to evaluate the performance of SLAM algorithms and provide a standardized database for research in the field of robot localization and navigation.

4.5 The Advantages of Redundant Line Feature Merging

This section provides a qualitative analysis of the redundant line deletion (merging) proposed in this paper. As mentioned earlier, the initial pose estimation in SLAM technology is calculated based on matched feature pairs. Therefore, the quality of the features used for matching and the accuracy of the matching significantly determine its performance. Hence, this section measures the advantage of the improved line features over the unimproved ones using the metric of matching accuracy. The accuracy metric is expressed as a percentage, calculated by subtracting the number of successfully matched line features from the total number of extracted line features, and then dividing the result by the total number of line features. In this paper, using this metric, we conducted experiments on line feature extraction and matching with 10 images as a group, and a total of 10 groups of images. With other experimental conditions remaining constant (such as lighting), the average accuracy of line feature matching before

improvement was 89.42%, while the average accuracy after improvement was 92.74%. This demonstrates the advantage of the proposed method.

4.6 Positioning Precision Assessment

To verify the performance of LR-SLAM, experiments were conducted on the EUROC and KITTI benchmark datasets, and compared with some SLAM systems such as VINS-Mono, ORB-SLAM3 Monocular, and PL-VIO. The following results were obtained (Tables 1, 2, 3, 4, and 5).

To validate the performance of LR-SLAM, we conducted experiments on the EUROC and KITTI benchmark datasets, comparing it with SLAM systems such as VINS-Mono, ORB-SLAM3 Monocular, and PL-VIO. The results obtained are presented in Tables 1 to 5. Tables 1 to 3 showcase the average Absolute Trajectory Error (ATE) test results for four representative sequences from the EUROC dataset. Table 4 displays the comparative results between LR-SLAM and other systems in the aforementioned tests. These results were calculated using Eqs. 11, 12, and 13. From the results, it is evident that LR-SLAM maintains a higher level of accuracy compared to VINS-Mono, ORB-SLAM3 Monocular, and PL-VIO. The reason why the method proposed in this

Fig. 6 (a) Compare the trajectory predictions of VINS-Mono (blue), ORB-SLAM3 monocular (green), PL-VIO (red), and LR-SLAM (purple) for sequence MH05 with the ground truth trajectory (brown). (b), (c) and (d) respectively show the comparisons between VINS-Mono, ORB-SLAM3, and PL-VIO (green) with LR-SLAM (blue) and the ground truth (brown)
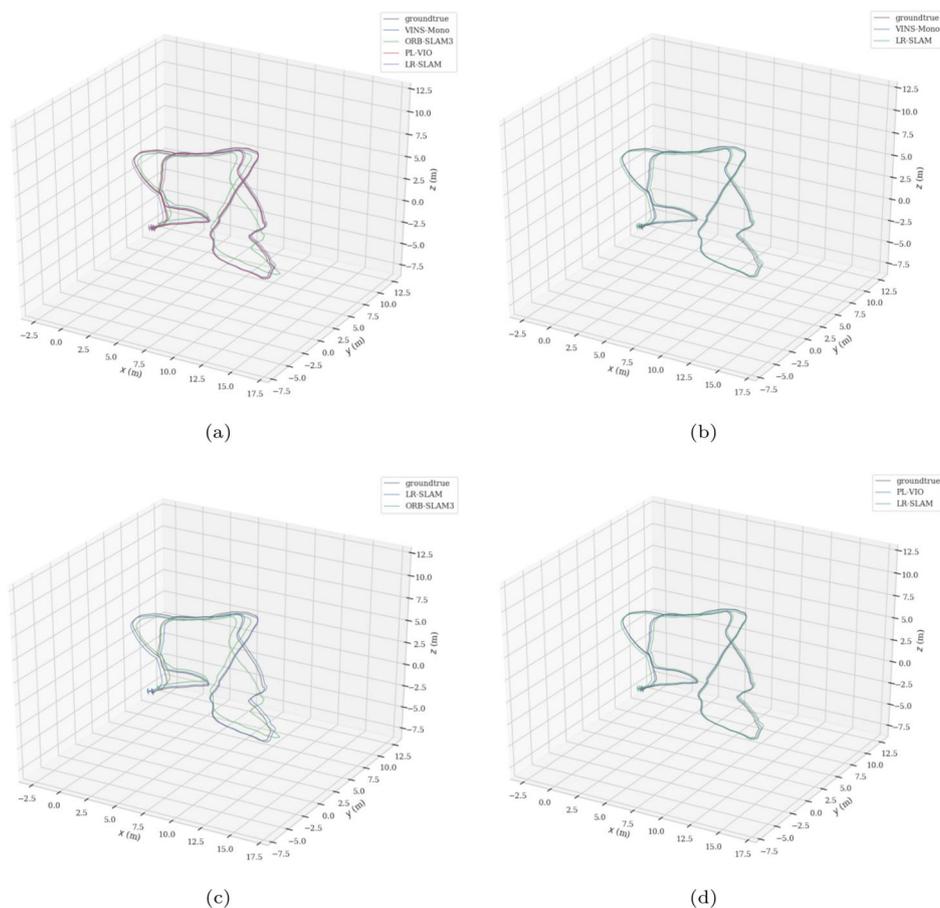
paper uses point features combined with line features to estimate pose is that line features can be easier to extract and have higher quality in certain scenarios compared to point features. For example, line features are relatively easy to extract on man-made structures and large objects, and they possess a higher level of abstraction compared to point features. Therefore, SLAM systems based on line features can perform better in certain scenarios than SLAM systems based on point features, such as the ORB-SLAM algorithm in the experiments of this section, which is a system based on point features and exhibits poorer pose estimation performance.

From Figs. 5 to 6, it can be seen that the total error of LR-SLAM is smaller compared to that of VINS-Mono, ORB-SLAM3 monocular, and PL-VIO, and it can be verified in Tables 1-5. In conclusion, the estimated trajectories of LR-SLAM are more consistent with the ground truth than those of VINS-Mono, ORB-SLAM3 monocular, and PL-VIO.

$$\frac{rmse_1 - rmse_2}{rmse_1} * 100, \tag{11}$$

$$\frac{std_1 - std_2}{std_1} * 100, \tag{12}$$

$$\frac{std_1 - std_2}{std_1} * 100, \tag{13}$$

where $rmse_1$ is the root-mean-square error of VINS-Mono, ORB-SLAM3 monocular and PL-VIO, $rmse_2$ is the root-mean-square error of this system, std_1 is the sum-of-squares error of VINS-Mono, ORB-SLAM3 Monocular and PL-VIO and std_2 is the sum-of-squares error of this system, $mean_1$ is the mean absolute error of VINS-Mono, ORB-SLAM3 monocular and PL-VIO, $mean_2$ is the mean absolute error of this system.

5 Conclusion

In this paper, a line feature fusion method is proposed with the aim of improving the accuracy of SLAM by this method. In addition to the line feature fusion method, sub-pixel point features and optional information matrix marginalization optimization methods are also utilized to enhance the positioning accuracy of monocular inertial navigation fused SLAM systems. Experimental results show that the line feature fusion method effectively mitigates the position estimation error due to repeated detection of the same lines in consecutive frames. With recent advances in the field of machine learning, parameterization, uncertainty, and multidimensional machine learning algorithms [28–30] can provide more effective learning and prediction capabilities when using limited features. These algorithms can be incorporated into SLAM algorithms that are tuned to specific

parameters, improving the algorithm's adaptability and generalization capabilities, and performing well in the face of missing or incomplete features.

Data Availability The data and code are not available in a public repository.

Declarations

Ethics Not applicable.

Conflict of Interests The authors declare no conflict of interest.

Consent for Publication All authors consent to publication.

Consent to Participate Not applicable.

References

1. Udugama, B.: Evolution of slam: toward the robust-perception of autonomy. **abs/2302.06365**, (2023). https://doi.org/10.48550/arXiv.2302.06365

2. Bala, J.A., Adeshina, S.A., Aibinu, A.M.: Advances in visual simultaneous localisation and mapping techniques for autonomous vehicles: a review. Sens. **22**(22), (2022). https://doi.org/10.3390/s22228943

3. Cheng, J., Zhang, L.Y., Chen, Q.H., Hu, X.R., Cai, J.C.: A review of visual slam methods for autonomous driving vehicles. Eng. Appl.

Artif. Intell. **114**, (2022). https://doi.org/10.1016/j.engappai.2022.104992

4. Wu, Y.H., Ta, X.X., Xiao, R.C., Wei, Y.G., An, D., Li, D.L.: Survey of underwater robot positioning navigation. Appl. Ocean Res. **90**, (2019). https://doi.org/10.1016/j.apor.2019.06.002

5. Dai, Y.: Research on robot positioning and navigation algorithm based on slam. Wireless Commun. Mobile Comput.. **2022**, (2022). https://doi.org/10.1155/2022/3340529

6. Sonugur, G.: A review of quadrotor uav: Control and slam methodologies ranging from conventional to innovative approaches. Robot. Auton. Syst. **161**, (2023). https://doi.org/10.1016/j.robot.2022.104342

7. Yin, Z., Wen, H., Nie, W., Zhou, M.: Localization of mobile robots based on depth camera. Remote Sens. **15**, 16 (2023). https://doi.org/10.3390/rs15164016

8. Mikulová, Z., Duchoň, F., Dekan, M., Babinec, A.: Localization of mobile robot using visual system. Int. J. Adv. Robot. Syst. **14**, 5 (2017). https://doi.org/10.1177/1729881417736085

9. Li, X., Gao, S., Yanqiang Yang, Y.Q., Liang, J.: The geometrical analysis of localization error characteristic in stereo vision systems. Rev. Sci. Instrum. **92**, (2021). https://doi.org/10.1063/5.0014379

10. Taihú, P., Thomas, F., Gastón, C., Pablo, D.C., Javier, C., Julio, J.B.: S-ptam: stereo parallel tracking and mapping. Robot. Auton. Syst. **93**, 27–42 (2017). https://doi.org/10.1016/j.robot.2017.03.019

11. Zhang, S.S., Zheng, L.Y., Tao, W.B.: Survey and evaluation of rgb-d slam. IEEE Access **9**, 21367–21387 (2021). https://doi.org/10.1109/access.2021.3053188

12. Raul, M., Juan, D.T.: Orb-slam2: an open-source slam system for monocular, stereo, and rgb-d cameras. IEEE Trans. Rob. **33**(5), 1255–1262 (2017). https://doi.org/10.1109/tro.2017.2705103

13. Qin, T., Li, P., Shen, S.: Vins-mono: a robust and versatile monocular visual-inertial state estimator. IEEE Trans. Rob. **34**(4), 1004–1020 (2018). https://doi.org/10.1109/TRO.2018.2853729

14. Ruben, G.O., Francisco-Angel, M., David, Z.N., Davide, S., Javier, G.J.: Pl-slam: a stereo slam system through the combination of points and line segments. IEEE Trans. Rob. **35**(3), 734–746 (2019). https://doi.org/10.1109/tro.2019.2899783

15. He, Y., Zhao, J., Guo, Y., He, W., Yuan, K.: Pl-vio: tightly-coupled monocular visual-inertial odometry using point and line features. Sens. **18**(4), (2018). https://doi.org/10.3390/s18041159

16. Fu, Q., Wang, J., Yu, H., Ali, I., Guo, F., Zhang, H.: Pl-vins: real-time monocular visual-inertial slam with point and line. arXiv. **no**(no), (2020). arXiv: 2009.07462

17. Yao, J., Zhang, P., Wang, Y., Luo, Z., Ren, X.: An adaptive uniform distribution orb based on improved quadtree. IEEE Access **7**(no), 143471–143478 (2019). https://doi.org/10.1109/ACCESS.2019.2940995

18. Campos, C., Elvira, R., Rodríguez, J.J.G., Montiel, J.M.M., Tardós, J.D.: Orb-slam3: an accurate open-source library for visual, visual–inertial, and multimap slam. IEEE Trans. Rob. **37**(6), 1874–1890 (2021). https://doi.org/10.1109/tro.2021.3075644

19. Liu, C.Y., Xu, J.S., Wang, F.: A review of keypoints' detection and feature description in image registration. Sci. Program. **2021**(no), (2021). https://doi.org/10.1155/2021/8509164

20. Ameer, N.O., Xia, Y.Q., Zhan, Y.F., Khurram, H., Iftikhar, A.K.: Systematic literature review on approaches of extracting image merits. Optik **271**(2), 5 (2022). https://doi.org/10.1016/j.ijleo.2022.170097

21. Dan X., Gong Q., Zhang M., Li T., Li G., Wang Y.: Chessboard corner detection based on edlines algorithm. Sens. **22**(9), (2022). https://doi.org/10.3390/s22093398

22. Liu, L., Zhang, T., Leighton, B.: Robust global structure from motion pipeline with parallax on manifold bundle adjustment and initialization. IEEE Robot. Autom. Lett. **4**(2), 2164–2171 (2019). https://doi.org/10.1109/lra.2019.2900756

23. Heo, S., Cha, J., Park, C.G.: Ekf-based visual inertial navigation using sliding window nonlinear optimization. IEEE Trans. Intell. Transp. Syst. **20**(7), 2470–2479 (2019). https://doi.org/10.1109/TITS.2018.2866637

24. Michael, B., Janosch, N., Pascal, G., Thomas, S., Joern, R., Sammy, O., Markus, W.A., Roland, S.: The euroc micro aerial vehicle datasets. The Int. J. Robot. Res. **35**(10), 1157–1163 (2016). https://doi.org/10.1177/0278364915620033

25. Geiger, A., Lenz, P., Stiller, C., Urtasun, R.: Vision meets robotics: The kitti dataset. The Int. J. Robot. Res. **32**(11), 1231–1237 (2013). https://doi.org/10.1177/0278364913491297

26. Amey, K.: Benchmarking and comparing popular visual slam algorithms. arXiv-CS-Robotics. **no**(no), (2018). arXiv:1811.09895

27. Seong, H.L., Javier, C.: What's wrong with the absolute trajectory error? arXiv-CS-Robotics. **no**(no), (2023). arXiv:2212.05376

28. Onder, T., Adem, P.: Linear and non-linear dynamics of the epidemics: system identification based parametric prediction models for the pandemic outbreaks. ISA Trans. **124**(no), 90–102 (2022). https://doi.org/10.1016/j.isatra.2021.08.008

29. Onder, T.: Graph theory based large-scale machine learning with multi-dimensional constrained optimization approaches for exact epidemiological modeling of pandemic diseases. IEEE Trans. Pattern Anal. Mach. Intell. **45**(8), 9836–9845 (2023). https://doi.org/10.1109/tpami.2023.3256421

30. Yan, J.J., Zheng, Y.B., Yang, J.Q., Lyudmila, M., Yuan, W.J., Gu, F.Q.: Plpf-vslam: An indoor visual slam with adaptive fusion of point-line-plane features. J. Field Robot. **41**(1), 50–67 (2024). https://doi.org/10.1002/rob.22242

Hao Jiang received his undergraduate degree from the College of Science and Technology of Three Gorges University in 2022. Currently, he is pursuing his master degree at the School of Information and Communication Engineering, Hainan University, Hainan, China. His research interests include computer vision, visual SLAM.

Naimeng Cang received the B.S. degree from Huaiyin Institute of Technology, Huaian, China, in 2018, and the M.S. degree from Shanghai Institute of Technology, Shanghai, China, in 2021. Currently, she is pursuing the Ph.D. degree in information and communication engineering at Hainan University, where her research focuses on intelligent robotics, image processing, pattern recognition, and multi-source information fusion.

Yuan Lin received his M.S. degree in Traffic and Transportation Engineering from Beihang University, Beijing, China, in 2022. He joined Hainan University in 2022. He is now a Ph. D student of Information and Communication Engineering in Hainan University. His research interests include path planning, path tracking, artifcial intelligence.

Dongsheng Guo received the B.S. degree in automation from Sun Yat-sen University, Guangzhou, China, in 2010, and the Ph.D. degree in communication and information systems from Sun Yat-sen University, Guangzhou, China, in 2015. Then, he joined Huaqiao as an Associate Professor. From 2018 to 2019, he was a Visiting Scholar with the National University of Singapore, Singapore. He is currently a Professor with the School of Information and Communication Engineer-

ing, Hainan University, Haikou, China, and also with the Engineering Research Center of Marine Intelligent Systems, Ministry of Education, China. His current research interests include robotics, neural networks, and numerical methods.

Weidong Zhang received his B.S., M.S., and Ph.D. degrees from Zhejiang University, China, in 1990, 1993, and 1996, respectively, and then worked as a Postdoctoral Fellow at Shanghai Jiao Tong University. He joined Shanghai Jiao Tong University in 1998 as an Associate Professor and has been a Full Professor since 1999. From 2003 to 2004 he worked at the University of Stuttgart, Germany, as an Alexander von Humboldt Fellow. He is a recipient of National Science Fund for Distinguished Young Scholars of China. In 2011 he was appointed Chair Professor at Shanghai Jiao Tong University. He is currently a Professor with the School of Information and Communication Engineering, Hainan University, Haikou, China, and also with the Engineering Research Center of Marine Intelligent Systems, Ministry of Education, China. His research interests include control theory, machine learning theory, and their applications in industry and autonomous systems. He is the author of more than 300 refereed papers and 1 book, and holds 32 patents.