Received 20 September 2022; revised 15 October 2022; accepted 22 October 2022. Date of publication 26 October 2022; date of current version 4 November 2022.

Digital Object Identifier 10.1109/OJCOMS.2022.3217147

A Real-time and Robust Monocular Visual Inertial SLAM System Based on Point and Line Features for Mobile Robots of Smart Cities Toward 6G

ZHENFEI KUANG[1], WEI WEI[1] (Member, IEEE), YIER YAN[1], JIE LI[1], GUANGMAN LU[1], YUYANG PENG[2] (Senior Member, IEEE), JUN LI[1] (Member, IEEE), AND WENLI SHANG[1] (Member, IEEE)

[1]School of Electronics and Communication Engineering, Guangzhou University, Guangzhou 510006, China

[2]School of Computer Science and Engineering, Macau University of Science and Technology, Macau, China

CORRESPONDING AUTHORS: W. SHANG and J. LI (e-mail: shangwl@gzhu.edu.cn; lijun52018@gzhu.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61905045 and Grant 62173101; in part by the Guangzhou Science and Technology Project under Grant 202102010501; and in part by the Open Research Project of Zhijiang Laboratory under Grant 2021KF0AB06.

(Zhenfei Kuang and Wei Wei contributed equally to this work.)

ABSTRACT Autonomous navigation of mobile robots in complex environments is challenging. Solving the problems of inaccuracy localization and frequent tracking losses of mobile robots in challenging scenes is beyond the power of point-based visual simultaneous localization and mapping (vSLAM). This paper proposes a real-time and robust point-line based monocular visual inertial SLAM (VINS) system for mobile robots of smart cities towards 6G. To extract robust line features for tracking in challenging scenes, EDLines with adaptive gamma correction is adopted to fast extract a larger ratio of long line features among all extracted line features. A real-time line feature matching approach is proposed to track the extracted line features between adjacent frames without the need of computing descriptors. Compared with LSD and KNN matching method based on LBD descriptors, the proposed method runs three times faster. Furthermore, a tightly coupled sensor fusion optimization framework is constructed for accurate state estimation, which contains point-line feature reprojection errors and IMU residuals. By evaluating on public benchmark datasets, our VINS system has high localization accuracy, real-time performance and robustness compared with other advanced SLAM systems. Our VINS system enables mobile robots to locate accurately in smart cities with complex environments.

INDEX TERMS SLAM, smart cities, mobile robots, sensor fusion, 6G.

I. INTRODUCTION

SMART cities are consisted by intelligent mobile robots, fully autonomous vehicles, ubiquitous Internet of Things, and big data. Knowing its location in an environment is a fundamental for a mobile robot autonomously executing tasks in smart cities. Simultaneous localization and mapping (SLAM) is a process by which a mobile robot builds a map of the surrounding environment while using the map to compute its location [1]. Nowadays, 5G wireless networks have been deployed widely to provide high-speed wireless communications with low latency and support basic autonomous systems [2]. However, it is debatable whether they can deliver high-level mobile robots and fully autonomous vehicles in smart cities [3]. With merits of higher data rates, lower latency and mobility of access in 6G wireless networks, the technology of SLAM applied in mobile robots and autonomous vehicles would drive the progress of smart cities [4], [5], [6], [7].

Visual SLAM (vSLAM) plays an extremely important role in the autonomous navigation [8], [9], [10], [11] of mobile robots in smart cities. In real-world environments of smart cities, there exist a large number of challenging scenes containing weak textures and motion blur caused by fast camera movements. Monocular vSLAM is not robust

enough for navigation in smart cities due to blurred images induced by fast motions and failure to align poses with gravity direction. Sensor fusion schemes using a combination of monocular cameras and inertial measurement unit (IMU) are proposed to compensate the above deficiencies. The fusion scheme is called monocular visual inertial odometry (VIO) [12] or monocular visual inertial SLAM (VINS) [13], [14], [15]. When tracking is lost caused by blurred images during fast motion, prediction of feature movements by IMU after pre-integration provides accurate initial values for tracking subsequent images. The gravity vector provided by the accelerometer of IMU transforms the camera coordinate to the world coordinate for camera poses. In addition, the zero bias of the IMU are corrected to effectively eliminate the cumulative drift of the IMU by combing rotations of image frames calculated by monocular vSLAM with extrinsic matrices from camera to IMU. Thus, VINS schemes are capable of improving localization accuracy and robustness of tracking.

It is referred above that IMU enables tracking subsequent images when tracking is lost. However, IMU is helpless when enough robust features can't be extracted in challenging scenes. Tracking success lies in extracting enough features and correctly matching them in real time. Point features are the most popular and common visual features for fast extraction and matching. In challenging scenes, only a small amount of point features can be extracted, which are insufficient for continuous tracking and accurate localization. Therefore, it is necessary to employ visual features with better robustness in these scenes. Line features that contain rich structural information and good invariabilities of light and rotation [16], are capable of improving the robustness and accuracy of tracking. However, extraction and tracking of line features require excessive computational resources and time consumption, leading to severe reduction of real-time performance.

To address the above problems, we propose a real-time and robust VINS system based on point-line features. It is implemented with the help of VINS-Mono [13] that is a state-of-the-art (SOTA) point-based VINS system. We propose a real-time line feature tracking process and then reconstruct the spatial lines in front-end. In back-end, we construct a new cost function by tightly coupling line reprojection errors, point reprojection errors and IMU residuals. Finally, accurate camera states are obtained by minimizing the cost function. Our main contributions are as follows:

1) We propose a line feature extraction method based on adaptive gamma correction and EDLines [17]. We compare the proposed method with LSD [18] and FLD [19] in terms of time consumption and the ratio of long line features among total quantity of line features. Experimental results show that the proposed method can fast extract a larger ratio of long line features among all line features, which is beneficial to the reduction of mis-matching rate and time consumption of line feature matching.
2) We propose a real-time line feature matching method based on the pyramidal Lucas-Kanade (LK) optical flow method [20]. The matched line features are used to reconstruct 3-D lines. Then, line reprojection errors are tightly coupled with point reprojection errors and IMU residuals to construct cost function for accurate state estimation.
3) We evaluate the VINS system on two public datasets of EuRoC MAV [21] and TUM-VI [22]. The results demonstrate that our VINS system performs well in terms of localization accuracy and real-time performance when compared with existing advanced technologies.

The rest of the paper is structured as follows. Section II describes the related work. Section III presents an overview of our VINS system. Section IV describes the methods for line feature extraction, matching, parameterized representation and adding line reprojection errors as a new constraint term to optimization framework. Section V provides the results of the experimental evaluation. Section VI gives conclusions for this paper.

II. RELATED WORK

A. FEATURE-BASED METHODS

1) POINT-BASED METHODS

At present, the classical and excellent point feature extraction algorithms are SIFT [23], SURF [24], ORB 0 and Shi-Tomasi [26]. There are an increasing number of vSLAM systems proposed using these algorithms. Among them, ORB-SLAM3 [14] and VINS-Mono are recognized as vSLAM benchmarks because they can be applied to a wide range of scenes and have good localization accuracy. However, in challenging scenes, such as motion blur and weak textures, the localization accuracy of point-based methods may be severely degraded and even tracking is lost.

2) POINT AND LINE-BASED METHODS

To ensure continuous tracking and improve the localization accuracy of vSLAM in challenging environments, it is important to introduce geometric features with geometric constraints such as lines and surfaces. The earlier proposed point-line based systems validate the improvement. PL-SVO [27], a semi-direct monocular visual odometer combining point and line segmentation, improves the performance of SVO [28] in low-texture environments. PL-StVO [29] introduces the probability of point-line based on PL-SVO to further improve the accuracy of camera state estimation and supports stereo vision. PL-SLAM [30] implements a more complete stereo visual SLAM process and achieves a more accurate loop closure accuracy based on PL-StVO, which uses ORB for the point feature extraction, LSD for the line feature extraction and LBD [31]. to describe the line features. In addition, for most point and line-based VIO/VINS such as PL-VIO [12], PL-VINS [15] and Trifo-VIO [32],

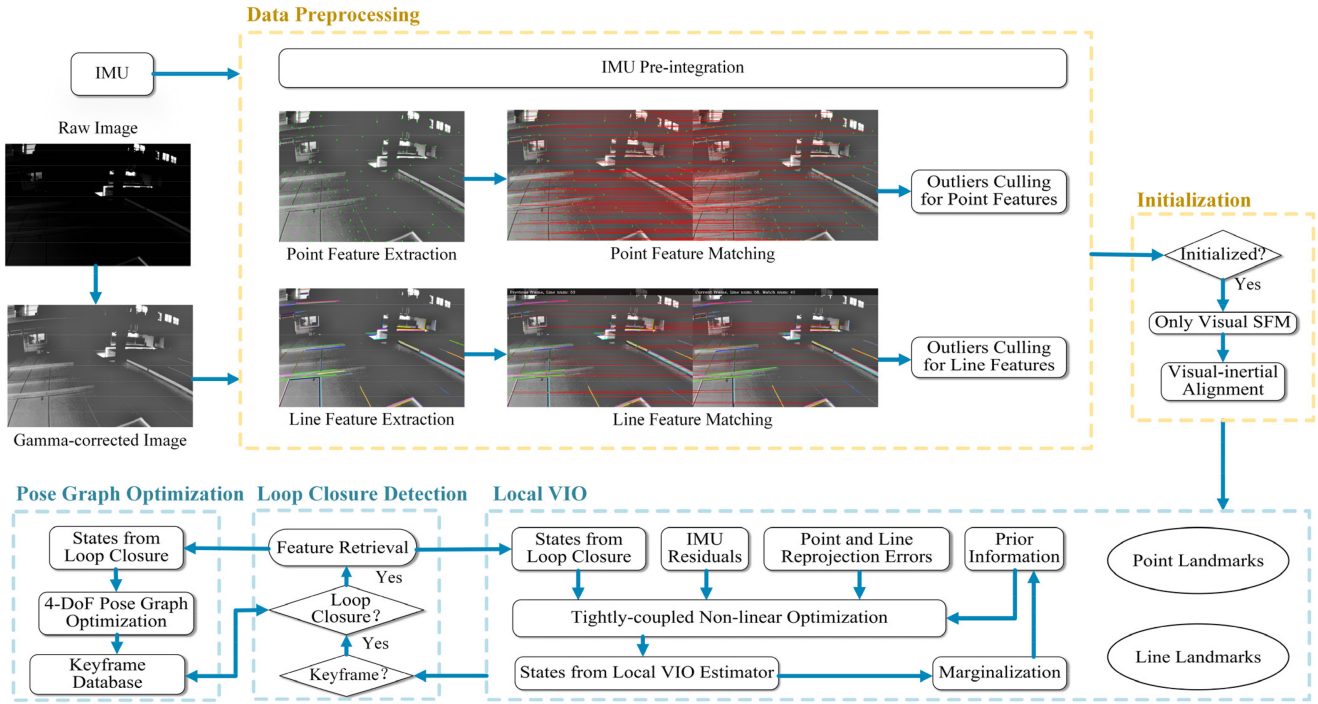


FIGURE 1. System overview.

line features are extracted by LSD, described by LBD, and then matched frame by frame using KNN [33]. However, both LSD extracting line features and LBD computing descriptors are quite time-consuming, which seriously affects the real-time performance of the system.

B. MULTI-SENSOR FUSION AND STATE ESTIMATION METHODS

The current multi-sensor fusion methods for visual information and inertial measurement information are divided into two categories: tightly coupled and loosely coupled. The loosely coupled algorithms use visual and IMU information as two independent state variables for their own state estimation, and then fuse the two state estimation results, as in MSF [34]. The tightly coupled algorithms use the raw data from visual and inertial measurements as the same state variable, and then obtain a globally consistent trajectory by means of state estimation, as in [35]. Although the tightly coupled algorithms are complex to solve, they can make full use of the sensor data and yield more accurate state estimation.

In the back-end, the state estimation methods are classified into optimization-based methods and filtering-based methods. The nonlinear optimization-based methods establish constraint relations between all state variables to be optimized and construct the objective function that is a least squares problem, and then solve it by bundle adjustment (BA). For filtering-based methods, the state of the current moment is only related to the state of the previous moment. The filtering-based methods mainly are based on extended Kalman filter (EKF) [36] and particle filter (PF) [37]. Although the computation of optimization-based methods is higher than that of filtering-based methods, higher accuracy can be obtained by optimization-based methods.

The study of state estimation is mainly using tightly coupled algorithms. The main systems of tightly coupled algorithms based on filtering methods are ROVIO [38], S-MSCKF [39] and MSCKF [40], all of which use Kalman filtering as a framework for improvement. The main systems of tightly coupled algorithms based on optimization methods are OKVIS [41] VINS-Mono, PL-VIO, and PL-VINS, all of which use sliding window for optimization. Our VINS system is an optimization-based tightly-coupled scheme, which contains prior information, IMU residuals and point-line reprojection error to construct the objective function and marginalizes the redundant information when a new frame is added. The optimization scheme is only to optimize the keyframes, while using the Ceres Solver [42] developed by Google to solve the nonlinear optimization problem.

III. SYSTEM OVERVIEW

In this paper, an overview diagram of our VINS system is shown in Fig. 1. It consists of five modules, which are data preprocessing, initialization and local VIO in front-end, loop closure detection and pose graph optimization in back-end.

A. FRONT-END

1) DATA PREPROCESSING

The input data consists of camera images and IMU measurement information. For image information, it is divided into

two parallel processes that are point feature tracking and line feature tracking. The point feature tracking process uses the Shi-Tomasi method to extract point features and the pyramidal LK optical flow method for point feature tracking. The line feature tracking process uses the proposed line extraction method to extract line features and the proposed line feature matching method for line feature tracking. For IMU measurement information, the IMU data are pre-integrated to obtain the pose, velocity, and rotation angle at the current moment. Meanwhile, it is calculated that the IMU pre-integration increments between adjacent frames, covariance matrix and Jacobi matrix, which will be used in the back-end optimization.

2) INITIALIZATION

Point and line landmarks are first recovered by triangulation based on the projection of point and line features under two camera frames. The spatial points and lines are represented by the inverse depth [43] and the Plücker coordinates [44], respectively. Based on these 3D features, visual-only SFM uses the PnP method [45] to estimate the poses of all frames in the sliding window. Then, visual SFM is aligned with IMU pre-integration to solve for the initialization parameters, including scale, gravity, velocity and bias.

3) LOCAL VIO

After initialization, the visual and inertial state variables are nonlinearly optimized in a sliding window of fixed size by minimizing the objective function with visual and IMU constraints. Other constraint terms in the sliding window include priori information generated from the marginalized frames and loop closure constraint generated when a loop closure is detected. In which, marginalization refers to removing the oldest frame or the sub-new frame when a new frame is added. Here, the marginalization is based on the Schur complement method [46]. The non-linear optimization of local VIO produces accurate states.

B. BACK-END

1) LOOP CLOSURE DETECTION

This module is designed to reduce the cumulative drift by finding the connection between the current frame and the candidate frames. BRIEF [47] is adopted to describe all FAST [48] corner points and calculate similarity scores of the current frame with all frames in the keyframe database. After temporal and spatial consistency checks, DBoW2 [49] returns candidate frames of loop closure detection. Among the candidate frames, all keyframes are obtained when the parallax between adjacent frames exceeds the threshold or when the number of features tracked is less than the threshold.

2) POSE GRAPH OPTIMIZATION

Pose graph optimization is activated when a loop closure is detected. It is added to the pose graph when a keyframe is marginalized from the sliding window. This keyframe is treated as a vertex of the pose graph, which establishes sequence edges with the previous five vertexes. If a closed-loop connection exists between this keyframe and other keyframes, linking them forms a closed-loop edge. After completing the construction of the pose graph, 4-DoF pose graph optimization is performed. Based on the optimization results, the past poses are updated by globally consistently configured. Global pose graph optimization produces more accurate poses.

IV. METHODOLOGY

In this paper, our VINS system is implemented with the help of VINS-Mono, in which line features are adopted to improve localization accuracy and robustness in challenging scenes.

Therefore, we elaborate line feature processes including extraction, matching, parameterized representation, reprojection errors and tightly coupling point-line reprojection errors with IMU residuals for optimization.

A. LINE FEATURE EXTRACTION

The proposed line feature extraction algorithm is based on the EDLines method and the adaptive gamma correction technique [50]. As shown in Figs. 2 and 3, EDLines with adaptive gamma correction plays an important role for line feature extraction. To reduce abundant extracted short features, we set a minimum length threshold of thirty-five. In particular, a line with length more than sixty is called a long line feature. Fig. 2(a) and (b) show the results of line feature extraction using the EDLines method on the images without gamma correction, where the total number of line features extracted and the number of long line features are too low. Fig. 3(a) and (b) show the results of line feature extraction using LSD and FLD on the same gamma-corrected image. LSD and FLD extract excessive short features among all extracted line features. Besides, the extraction of LSD takes a very long time. Fig. 2(c) and Fig. 3(c) show the results of the proposed line feature extraction method, and the method fast extracts a large ratio of long line features and small amount of short line features. The experimental results show that the proposed line feature extraction method has strong capability for extracting long line features and high real-time performance. The following is a validation of the role of the adaptive gamma correction method and the EDLines method for line feature extraction.

1) ADAPTIVE GAMMA CORRECTION

In VINS-Mono, only contrast limited adaptive histogram equalization (CLAHE) is used to enhance the contrast of the input images. As shown in Fig. 2, CLAHE is used to enhance image contrast rather than image luminance. As a result, limited number of line features are extracted and most of them are short line features. Similarly, only a limited number of point features can be extracted

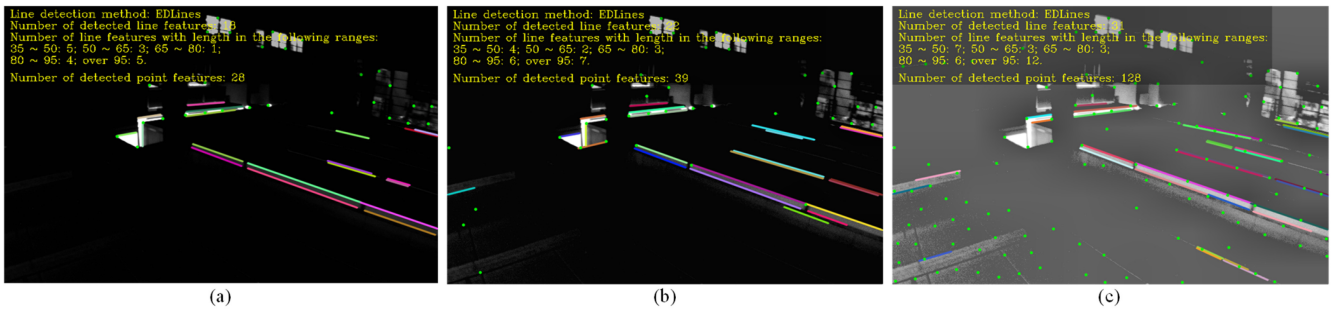


FIGURE 2. Visual feature extraction. (a), (b) and (c) are the results of extracting point-line features on the raw image, the image of CLAHE, and the gamma-corrected image, respectively. The detection results are noted in yellow font in the upper left corner of each image.

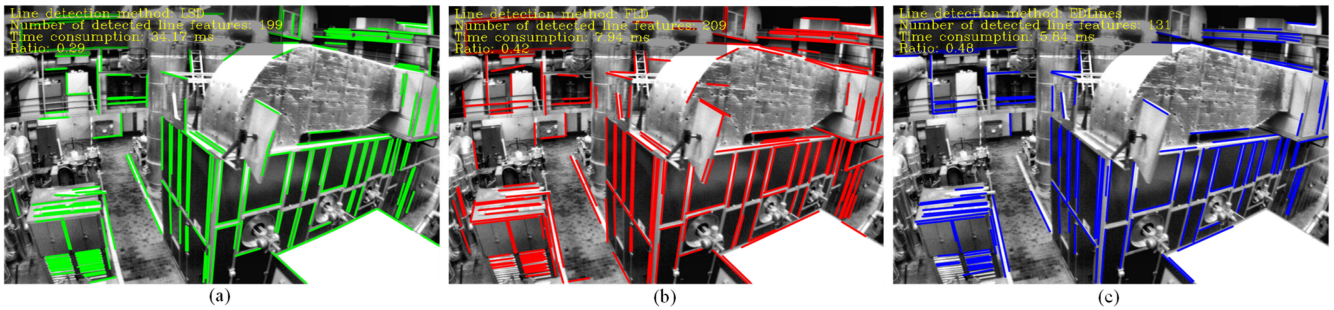


FIGURE 3. A comparison of the three algorithms of line feature detection on a single image frame. (a), (b) and (c) are the results of line feature detection by LSD, FLD and EDLines on the same gamma-corrected image, respectively. For intuitive comparison, the line features extracted by the three algorithms are drawn in green, red and blue, respectively. The detection method, total number of extracted line features, time consumption and ratio are noted in yellow font in the upper left corner of each image.

in the low-light images. Adaptive gamma correction can dynamically adjust the luminance values of images. The gamma value of adaptive gamma correction is determined by the luminance of images. For low-light images, the gamma value is always less than 1. A lower image luminance results in a smaller gamma value and a higher luminance improvement.

2) THE EDLINES LINE DETECTION METHOD

We compare three algorithms of LSD, FLD, and EDLines, which are at present most widely used in vSLAM for line feature extraction. As shown in Fig. 3 and Fig. 4, high "Ratio" of long line features among all extracted line features indicates that the extraction result for a certain frame is superior. The robustness of line feature extraction algorithm is reflected in a large ratio of long line features and high real-time performance, which facilitates the reduction of mismatch rate and time consumption for line feature matching.

As shown in Fig. 3 and Fig. 4, we compare the above three methods in terms of time consumption, ratio of long line features and proper quantity testing on a single frame and a whole sequence, respectively. The results show that while time consumption of EDLines is comparable to that of FLD, the ratio of long line features is higher than the other two algorithms. In addition, the total quantity of extracted line features by EDLines is not excessive.

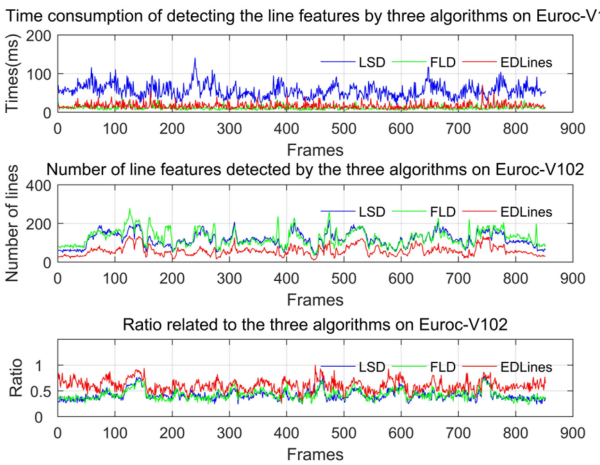


FIGURE 4. A comparison of the three algorithms of line feature detection on V1_02_medium of the EuRoC MAV dataset.

B. LINE FEATURE MATCHING

Line feature matching usually requires line descriptors calculated in advance from two adjacent frames as input. Most VINS systems use LBD to describe line features, but LBD is very complex and time-consuming. Therefore, we propose a line feature matching method based on the pyramidal LK optical flow method. The method requires no computation of descriptors and significantly improves the speed of line feature tracking.

Algorithm 1 The Proposed Line Feature Matching Method

Input: I_c, I_p, $L_c = \{l_i^c | l_i^c = (s_i^c, e_i^c),\ 0 \le i \le M_c\}$, $L_p = \{l_k^p | l_k^p = (s_k^p, e_k^p),\ 0 \le k \le M_p\}$, $\mathbf{n}_i^c$, $\mathbf{v}_i^c$, $\triangle s$, $d_{threshold}$, W
Output: l_i^c and $l_{match(i)}^p$ match each other
for $i \leftarrow 0$ **to** M_c **do**
  for s_i^c **to** e_i^c **do**
    $T^c = \{t_{i,a}^c | 0 \le a \le N_i^t\} \leftarrow \triangle s, \mathbf{v}_i^c$
    $M^p = \{m_{i,j}^p | 0 \le j \le N_i^m \le N_i^t\} \leftarrow T^c$, $\mathbf{n}_i^c$, the pyramidal LK optical flow method
    for $j \leftarrow 0$ **to** N_i^m **do**
      $D_{i,j} = \{d_{i,j,k} \ge 0 | 0 \le k \le M_p\} \leftarrow M^p, L_p$
      $d_{min(i,j)} \leftarrow min(D_{i,j})$
      if $d_{min(i,j)} < d_{threshold}$ **then**
        $m_{i,j}^p$ is on $l_{nearest(i,j)}^p$
        $t_{i,a}^c$ matched with $l_{nearest(i,j)}^p$
        if $l_{match(i)}^p$ and $l_{nearest(i,j)}^p$ as the same line **then**
          $Z_i^c ++$
        end
      end
    end
  if $Z_i^c / N_i^t > W$ **then**
    l_i^c and $l_{match(i)}^p$ match each other
  end
  end
end

LBD is an approach used to describe line features, which can describe local appearance of lines well for matching. LBD adopts the statistics of pixel gradients of all points in line support region (LSR) and calculates the mean vector and standard deviation of the statistics as descriptors, and this process is complex and quite time-consuming to complete. Pyramid LK optical flow tracking is an approach to track feature points, which is implemented through pyramid building, tracking, and iteration. Because of its simple steps and sparse optical flow tracking with fast speed, pyramidal LK optical flow tracking has good real-time performance. For matching, the traditional KNN matching method compares the similarity of line features of two adjacent frames after LBD description. The proposed line feature matching method is to find correspondences of points on line features of two adjacent frames. The proposed line feature matching method is shown in Algorithm 1 and described in detail as follows.

For two adjacent frames I_c and I_p, EDLines with adaptive gamma correction extract line features for each of them, represented by the sets $L_c = \{l_i^c | l_i^c = (s_i^c, e_i^c),\ 0 \le i \le M_c\}$ and $L_p = \{l_k^p | l_k^p = (s_k^p, e_k^p),\ 0 \le k \le M_p\}$, where $s, e \in \mathbb{R}^2$ are the two endpoints of the line segment l. For each line segment l_i^c in the current frame I_c, its direction vector $\mathbf{v}_i^c$ is calculated by its two endpoints. The normal vector $\mathbf{n}_i^c$ can be obtained by the orthogonal constraint $\mathbf{n}_i^{cT}\mathbf{v}_i^c = 0$. For line l_i^c, we let s_i^c, e_i^c, $\mathbf{v}_i^c$ and $\triangle s$ (default is 10) as start point, end point, direction and steps, respectively. From s_i^c to e_i^c, a set of points on l_i^c is selected for line feature matching. We call the selected points "**Tagged points**", denoted by the set of points as $T^c = \{t_{i,a}^c | 0 \le a \le N_i^t\}$, where N_i^t denotes the maximum number of points selected on l_i^c.

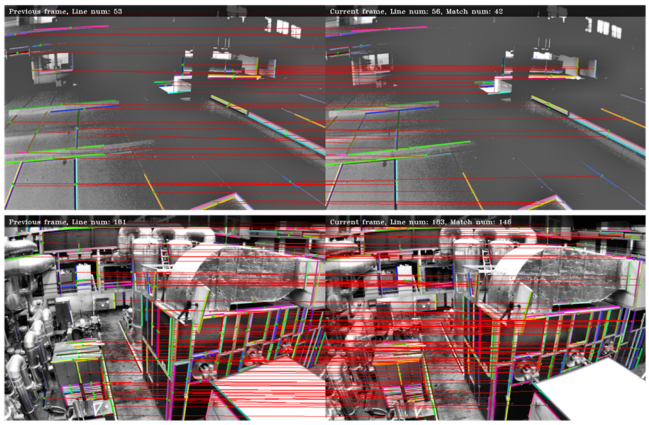


FIGURE 5. Results of line feature tracking with the proposed line feature matching method.

For points in T_c, point optical flow tracking is performed by using the pyramidal LK optical flow method to find the matched points in the previous frame I_p. The matched points are denoted by the set $M^p = \{m_{i,j}^p | 0 \le j \le N_i^m \le N_i^t\}$, where N_i^m denotes the maximum number of points matched by the "**Tagged points**" on l_i^c.

In the previous frame I_p, we calculate the shortest distance from each $m_{i,j}^p$ to all line segments in L_p, denoted by the set $D_{i,j} = \{d_{i,j,k} \ge 0 | 0 \le k \le M_p\}$. We compare all elements in $D_{i,j}$ to get the shortest distance $d_{min(\ i,j)}$. If $d_{min(\ i,j)}$ is not bigger than $d_{threshold}$ (default is 1), we consider that the matching point $m_{i,j}^p$ is on the corresponding line segment $l_{nearest(i,j)}^p$. We also consider the corresponding "**Tagged points**" $t_{i,a}^c$ matched with the line $l_{nearest(i,j)}^p$.

For line segment l_i^c, suppose there are Z_i^c "**Tagged points**" matching the same line segment $l_{match(i)}^p$. If $Z_i^c / N_i^t > W$ (we set it to 0.5), then we consider that l_i^c and $l_{match(i)}^p$ match each other. The results of line feature matching are shown in Fig. 5, where we connect midpoints of the matched line pairs in red lines to indicate successful match.

C. SPATIAL LINE REPRESENTATION

In point line-based vSLAM, spatial lines need to be reconstructed and used to optimization in back-end. In two processes, the Plücker coordinates representation is used for triangulation and the orthogonal representation is used for optimization.

1) PLÜCKER COORDINATES REPRESENTATION

As shown in Fig. 6(a), give the two endpoints s and e of spatial line segment $\boldsymbol{l}$, the Plücker coordinates of spatial line $\boldsymbol{\mathcal{L}}$ can be described as

$$\boldsymbol{\mathcal{L}} = \begin{bmatrix} s \times e \\ ms - ne \end{bmatrix} = \begin{bmatrix} \mathbf{n} \\ \mathbf{v} \end{bmatrix} \in \mathbb{R}^6 \tag{1}$$

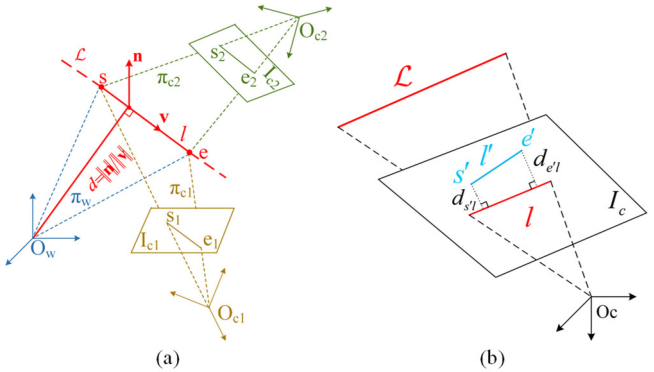


FIGURE 6. (a) Geometric change of lines. $(\cdot)_w$, $(\cdot)_{c_1}$ and $(\cdot)_{c_2}$ are world frame and two camera frames, respectively. I_{c_1} and I_{c_2} are the normalized planes of the two camera frames, respectively. (b) Reprojection error model of line features. I_c is camera imaging plane.

where $\mathbf{n} \in \mathbb{R}^3$ is the normal vector of the plane π_w determined by the world frame origin and the spatial line $\boldsymbol{\mathcal{L}}$, $\mathbf{v} \in \mathbb{R}^3$ is the direction vector of the line $\boldsymbol{\mathcal{L}}$, and both of them satisfy the orthogonal constraint $\mathbf{n}^T\mathbf{v} = 0$. m and n are two non-zero constants.

Suppose the line segment $\boldsymbol{l}$ is observed by two camera frames c_1 and c_2. The line segment $\boldsymbol{l}$ is projected onto the normalized plane I_{c_1}, and the projection points of its two endpoints are $s_1 \in \mathbb{R}^3$ and $e_1 \in \mathbb{R}^3$. Give the coordinates $\mathrm{O}_{c_1} = (x_{c_1}, y_{c_1}, z_{c_1})^T$ of the origin of c_1, the coordinates of plane $\pi_{c_1} = \left(\pi_1^{c_1}, \pi_2^{c_1}, \pi_3^{c_1}, \pi_4^{c_1}\right)$ can be obtained by

$$\begin{cases} [\pi_1^{c_1}, \pi_2^{c_1}, \pi_3^{c_1}]^T = [s_1]_\times e_1, \\ \pi_4^{c_1} = \pi_1^{c_1} x_{c_1} + \pi_2^{c_1} y_{c_1} + \pi_3^{c_1} z_{c_1}, \end{cases} \tag{2}$$

where $[\,\cdot\,]_\times$ denotes the skew-symmetric matrix of a three-dimensional vector. Similarly, the coordinates of plane $\pi_{c_2} = \left(\pi_1^{c_2}, \pi_2^{c_2}, \pi_3^{c_2}, \pi_4^{c_2}\right)$ can be obtained. According to [51], the dual Plücker matrix $\mathbf{L}^*$ can be represented as

$$\mathbf{L}^* = \begin{bmatrix} [\mathbf{n}]_\times & \mathbf{v} \\ -\mathbf{v}^T & 0 \end{bmatrix} = \pi_{c_1}\pi_{c_2}^T - \pi_{c_2}\pi_{c_1}^T. \tag{3}$$

$\mathbf{L}^*$ is a skew-symmetric matrix with six non-zero elements. Therefore, comparing the 4-DoF of a spatial line, both the Plücker coordinates and the Plücker matrix are over-parameterized representations.

2) ORTHONORMAL REPRESENTATION

In order to minimize computational resource and improve convergence in back-end nonlinear optimization, we use the four-parameter orthonormal minimal representation proposed by Bartoli and Sturm [44]. The orthonormal representation $(\mathbf{U}, \mathbf{W}) \in SO(3)\times SO(2)$ of the spatial line $\mathcal{L}$ can be obtained by using the QR decomposition on the matrix $[\mathbf{n}|\mathbf{v}]$:

$$QR([\mathbf{n} \mid \mathbf{v}]) = \mathbf{U}\begin{bmatrix} w_1 & 0 \\ 0 & w_2 \\ 0 & 0 \end{bmatrix}, \text{ set } \mathbf{W} = \begin{bmatrix} w_1 & -w_2 \\ w_2 & w_1 \end{bmatrix}, \tag{4}$$

where w_1 and w_2 are greater than zero. In addition, the transformation from the Plücker coordinates to the orthonormal representation can also be obtained by

$$[\mathbf{n} \mid \mathbf{v}] = \begin{bmatrix} \frac{\mathbf{n}}{\|\mathbf{n}\|} & \frac{\mathbf{v}}{\|\mathbf{v}\|} & \frac{\mathbf{n}\times\mathbf{v}}{\|\mathbf{n}\times\mathbf{v}\|} \end{bmatrix}\begin{bmatrix} \|\mathbf{n}\| & 0 \\ 0 & \|\mathbf{v}\| \\ 0 & 0 \end{bmatrix}. \tag{5}$$

Let $\mathbf{R}(\boldsymbol{\theta}) = \mathbf{U}, \mathbf{R}(\theta) = \mathbf{W}$, according to (4) and (5), we get

$$\begin{aligned} \mathbf{R}(\boldsymbol{\theta}) = \mathbf{U} &= \begin{bmatrix} \frac{\mathbf{n}}{\|\mathbf{n}\|} & \frac{\mathbf{v}}{\|\mathbf{v}\|} & \frac{\mathbf{n}\times\mathbf{v}}{\|\mathbf{n}\times\mathbf{v}\|} \end{bmatrix}, \\ \mathbf{R}(\theta) = \mathbf{W} &= \begin{bmatrix} cos(\,\theta) & -sin(\,\theta) \\ sin(\,\theta) & cos(\,\theta) \end{bmatrix} = \begin{bmatrix} w_1 & -w_2 \\ w_2 & w_1 \end{bmatrix} \\ &= \frac{1}{\sqrt{\|\mathbf{n}\|^2 + \|\mathbf{v}\|^2}}\begin{bmatrix} \|\mathbf{n}\| & -\|\mathbf{v}\| \\ \|\mathbf{v}\| & \|\mathbf{n}\| \end{bmatrix}. \end{aligned} \tag{6}$$

$\mathbf{W}$ contains the information about the distance from the origin O_w to the line $\boldsymbol{\mathcal{L}}$. Because this distance can be obtained and represented by $d = \|\mathbf{n}\|/\|\mathbf{v}\| = w_1/w_2$. It follows that the scalar $\theta \in (0, \pi/2)$ can be used to adjust the distance d. In addition, $\boldsymbol{\theta} \in \mathbb{R}^3$ can be used to regulate the rotation of a spatial line around the three axes of x-y-z at a constant distance d. Therefore, we can define the orthonormal representation in terms of a four-parameter vector:

$$\boldsymbol{\mathcal{O}} = (\boldsymbol{\theta}^T, \theta)^T. \tag{7}$$

Similarly, given the orthonormal representation $(\mathbf{U}, \mathbf{W})$, the Plücker coordinates can be obtained by the transformation formula:

$$\boldsymbol{\mathcal{L}} = \begin{bmatrix} w_1\mathbf{u}_1 \\ w_2\mathbf{u}_2 \end{bmatrix}, \tag{8}$$

where w_1, w_2, $\mathbf{u}_1$ and $\mathbf{u}_2$ can be obtained from (6) and $\mathbf{u}_i$ is the ith column of $\mathbf{U}$.

D. LINE FEATURE REPROJECTION ERROR

The Plücker coordinates can be used for linear coordinates transformation and reprojection representation. Spatial line transformation from the world frame to the camera frame can be achieved by the Plücker coordinates transformation

$$\mathbf{L}_c = \begin{bmatrix} \mathbf{n}_c \\ \mathbf{v}_c \end{bmatrix} = \begin{bmatrix} \mathbf{R}_{cw} & [\mathbf{t}_{cw}]_\times\mathbf{R}_{cw} \\ \mathbf{0} & \mathbf{R}_{cw} \end{bmatrix}\begin{bmatrix} \mathbf{n}_w \\ \mathbf{v}_w \end{bmatrix}, \tag{9}$$

where $\mathbf{R}_{cw} \in SO(3)$ is the rotation matrix and $\mathbf{t}_{cw} \in \mathbb{R}^3$ is the translation vector.

As shown in Fig. 6(b), the projection line $\boldsymbol{l}$ obtained by projecting the spatial line $\boldsymbol{\mathcal{L}}_c$ of the camera frame onto camera imaging plane can be represented as

$$\boldsymbol{l} = [l_1, l_2, l_3]^T = \mathbf{K}\mathbf{n}_c, \tag{10}$$

where $\mathbf{K}$ is the intrinsic matrix of camera. The reprojection error model of the line is determined by modeling the vertical distance from the endpoints (s', e') of the matched line $\boldsymbol{l}'$ to the projected line $\boldsymbol{l}$:

$$r_{\boldsymbol{\mathcal{L}}}(\hat{\boldsymbol{z}}_j^{c_i}, \boldsymbol{\chi}) = [d_{s'\boldsymbol{l}}, d_{e'\boldsymbol{l}}]^T = \left[\frac{s'\boldsymbol{l}}{\sqrt{l_1^2 + l_2^2}}, \frac{e'\boldsymbol{l}}{\sqrt{l_1^2 + l_2^2}}\right]^T, \tag{11}$$

where $\hat{z}_j^{c_i}$ is the observation of the ith camera frame c_i to the jth spatial line $\mathcal{L}_j$ and χ is a full-state vector.

E. SLIDING WINDOW OPTIMIZATION MODEL

The full state vector χ in the sliding window is defined as

$$\begin{aligned}\chi &= [\boldsymbol{x}_1, \cdots, \boldsymbol{x}_n, \lambda_1, \cdots \lambda_m, \mathcal{O}_1, \cdots, \mathcal{O}_l]^T, \\ \boldsymbol{x}_i &= \left[\boldsymbol{p}_{b_i}^w, \boldsymbol{v}_{b_i}^w, \boldsymbol{q}_{b_i}^w, \boldsymbol{b}_a^b, \boldsymbol{b}_g^b\right]^T, i \in [1,n],\end{aligned} \tag{12}$$

where $\boldsymbol{x}_i$ is the state vector of the IMU at the time of shooting the ith camera frame. It contains the position, velocity, and direction of the IMU in the world frame and the accelerometer bias and gyroscope bias in the IMU body frame. n, m and l denote the total number of keyframes, the number of feature points and the number of line features in the sliding window, respectively. λ_j is the inverse depth of the jth feature point from its first observed keyframe and used to parameterize that point. $\mathcal{O}_k$ is the orthonormal representation of the kth line feature, which is used to parameterize the line.

Based on the cost function of the back-end visual inertial BA of VINS-Mono, we introduce the reprojection errors of line features and modify the objective function in the sliding window as

$$\begin{aligned}\boldsymbol{C} = \min_{\chi}\Bigg\{ &\left\|\boldsymbol{r}_p - \boldsymbol{H}_p\chi\right\|^2 + \sum_{k\in B}\left\|\boldsymbol{r}_B(\hat{\boldsymbol{z}}_{b_{k+1}}^{b_k}, \chi)\right\|_{\boldsymbol{P}_{b_{k+1}}^{b_k}}^2 + \\ &\sum_{(l,j)\in C}\rho\left(\left\|\boldsymbol{r}_C(\hat{\boldsymbol{z}}_l^{c_j}, \chi)\right\|_{\boldsymbol{P}_l^{c_j}}^2\right) + \\ &\sum_{(i,k)\in \mathcal{L}}\rho\left(\left\|\boldsymbol{r}_{\mathcal{L}}(\hat{\boldsymbol{z}}_i^{c_k}, \chi)\right\|_{\boldsymbol{P}_i^{c_k}}^2\right) + \\ &\sum_{(l,v)\in LP}\rho\left(\left\|\boldsymbol{r}_C(\hat{\boldsymbol{z}}_l^{v}, \chi, \hat{\boldsymbol{q}}_v^w, \hat{\boldsymbol{p}}_v^w)\right\|_{\boldsymbol{P}_l^{c_v}}^2\right)\Bigg\},\end{aligned} \tag{13}$$

where $\{\boldsymbol{r}_p, \boldsymbol{H}_p\}$ is the priori information obtained by preserving the states when marginalizing old frames. $\boldsymbol{r}_B(\hat{\boldsymbol{z}}_{b_{k+1}}^{b_k}, \chi)$ is the IMU measurement residual. $\boldsymbol{r}_C(\hat{\boldsymbol{z}}_l^{c_j}, \chi)$ and $\boldsymbol{r}_{\mathcal{L}}(\hat{\boldsymbol{z}}_i^{c_k}, \chi)$ are the reprojection errors of the point features and line features, respectively. $\boldsymbol{r}_C(\hat{\boldsymbol{z}}_l^{v}, \chi, \hat{\boldsymbol{q}}_v^w, \hat{\boldsymbol{p}}_v^w)$ is the loop closure constraint added to the sliding window when a loop closure is detected, where LP is the set of observations of features retrieved in a loop closure frame. (l, v) denotes the lth feature point observed in the vth loop closure frame. $(\hat{\boldsymbol{q}}_v^w, \hat{\boldsymbol{p}}_v^w)$ is the pose of the loop closure frame. $\rho(\cdot)$ is the Huber norm [52] that is a loss function, which is defined as

$$\rho(s) = \begin{cases} s & s \le 1, \\ 2\sqrt{s} - 1 & s > 1. \end{cases} \tag{14}$$

which is used to obtain a higher optimization accuracy by suppressing the effect of noise.

V. EXPERIMENTS

To evaluate the trajectory accuracy and real-time performance of the proposed system, we choose to perform extensive experiments on the public EuRoC MAV dataset and the public TUM-VI dataset. The EuRoC MAV dataset consists of simultaneous pinhole stereo images, IMU measurement information and ground truth. All the data are collected by the Swiss Federal Institute of Technology Zurich using small UAVs in three indoor scenes including Machine Hall and Vicon Room 1, 2. The difference is that the stereo images of the TUM-VI dataset are collected by a fish-eye camera, which is a novel VI dataset proposed by the Technical University of Munich, Germany. The TUM-VI dataset contains several scenes, such as rooms, corridors, and outdoors.

After obtaining trajectories of the sequences of two datasets, we evaluate the performance of the proposed system using EVO tool. EVO is evaluating odometry or SLAM algorithms, which includes data evaluation and visualization functions. All experiments were performed on PC with AMD Ryzen 5 PRO @3.70 GHz CPU, 16GB RAM and implemented based on Ubuntu 18.04 and ROS Melodic.

A. TRAJECTORY ACCURACY

In this section, the purpose is to verify the ability of our VINS system to estimate the camera poses by evaluating trajectory accuracy. We select some advanced VIO systems and VINS systems to compare with our proposed system. As shown in Table 1, ROVIO, PL-VIO and OURS(w/o) are VIO systems and PL-VINS, VINS-Mono and OURS are VINS systems. Notice that VINS is a SLAM system which is generated by adding a loop closure module to VIO. Similarly, PL-VIO and PL-VINS are also obtained by introducing line features with the help of VINS-Mono. However, the different methods of line feature extraction, description and matching will cause a large gap in the trajectory accuracy and real-time performance, as can be seen from the following results.

Table 1 shows the comparison of the localization accuracy of these systems mentioned above by evaluating the root-mean-squared error (RMSE) of the absolute trajectory error (ATE). The "×" indicates that the system tracks abnormally on that sequence, for example, only partial trajectories are obtained or the error of the complete trajectory is quite large. The results show that our VINS system has the highest accuracy among all compared systems on most sequences of the EuRoC MAV dataset. And our system without loop closure OURS(w/o) has the lowest error in the comparison of VIO systems. In particular, the advantage is more obvious in the Vicon Room sequences that have weak textures, motion blur and fast motions. This indicates that our VINS system is robust when running in such challenging scenes, as will be described in detail in Section V-B.

As shown in Fig. 7, we compare the distances between the trajectories of the three VINS systems on sequences of V1_02_medium and V1_03_difficult with the ground truth, and the results in the XY plane. Although these trajectories are very similar, it can be seen that the trajectories obtained by our VINS system have the closest distance to the ground

TABLE 1. Localization accuracy with several VIO and VINS systems tested on the EuRoC MAV dataset. The RMSE of the absolute trajectory error (m) is used for evaluation.

	ROVIO	PL-VIO	PL-VINS	VINS-Mono	OURS	OURS(w/o)
MH_01_easy	0.2773	0.1358	0.0761	0.0844	**0.0719**	0.1917
MH_02_easy	×	0.1418	0.0624	**0.0558**	0.0755	0.1939
MH_03_medium	0.4453	0.2646	**0.0627**	0.0797	0.0651	0.2059
MH_04_difficult	0.7907	0.3633	0.1009	0.1341	**0.0990**	0.2685
MH_05_difficult	1.0708	0.3068	0.1550	0.1648	**0.1318**	0.2795
V1_01_easy	0.1594	0.0827	0.0446	0.0552	**0.0430**	0.0806
V1_02_medium	0.1315	×	0.0612	0.0635	**0.0398**	0.1079
V1_03_difficult	0.1984	0.2017	0.1824	0.2011	**0.1006**	0.1115
V2_01_easy	0.2416	0.0876	0.0601	0.0966	**0.0512**	0.0741
V2_02_medium	0.4176	0.1348	0.0857	0.1304	**0.0764**	0.1095
V2_03_difficult	0.2097	0.3004	**0.1360**	0.2175	0.1397	0.2615
Mean	0.3942	0.2020	0.0934	0.1166	**0.0813**	0.1713

* The best result or the smallest error is **bold**.
* The symbol "×" indicates that the system tracks abnormally on the sequence and the evaluated result is not indicative, so it is discarded.

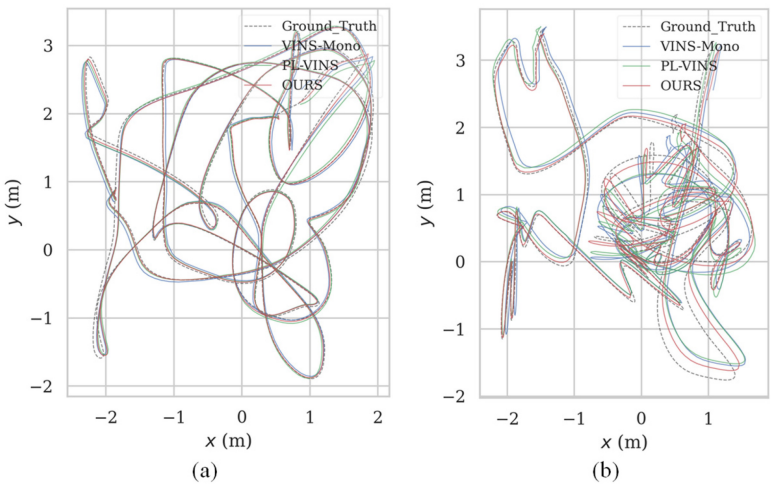


FIGURE 7. A comparison of the trajectories obtained by VINS-Mono, PL-VINS and OURS on (a) V1_02_medium and (b) V1_03_difficult of the EuRoC MAV dataset with the ground truth.

truth. From the above experiments on the EuRoC MAV dataset, our VINS system has high localization accuracy.

B. ROBUSTNESS TESTING IN CHALLENGING SCENES

In this section, we evaluate the robustness of our VINS system by evaluating the RMSE of trajectories obtained on sequences with challenging environmental factors. The localization accuracy of the three VINS systems is the highest as seen in Table 1, so we only compare their robust performance in this section. In Table 2, the RMSE of the ATE on the EuRoC MAV dataset is derived from Table 1. The RMSE of VINS-Mono on the TUM-VI dataset is derived from [22]. The RMSEs of both PL-VINS and our VINS system are evaluated on the same PC.

As shown in Table 2, the experimental results of the robustness test indicate that the point-line based system is more applicable than the point-based system in challenging scenes as seen from the apparent improvement in localization accuracy. Moreover, the proposed system is able to utilize robust point-line features in challenging scenes so as to improve the localization accuracy.

C. REAL-TIME PERFORMANCE

As mentioned above, PL-VIO, PL-VINS and our VINS system are obtained by introducing line features with the help of VINS-Mono. The purpose of this section is to the time consumption of the three systems for line feature extraction, line feature description and matching and the whole line feature tracking process, respectively.

For line feature extraction, PL-VIO and PL-VINS use the traditional LSD method and the improved LSD method integrated by OPENCV, respectively. Similarly, PL-VIO and PL-VINS use the traditional KNN matching method based on the LBD description. Our VINS system uses the proposed

TABLE 2. Robustness testing of our VINS system in challenging scenes of the EuRoC MAV dataset sequences and the TUM-VI dataset sequences. The RMSE of the ATE (m) is used for evaluation.

Dataset	Sequence	VINS-Mono	PL-VINS	OURS	Improvement	Challenging scenes
EuRoC MAV	MH_05_difficult	0.16	0.16	**0.13**	19%	Partly dark scene
	V1_02_medium	0.06	0.06	**0.04**	33%	Weak texture, Fast motion,
	V1_03_difficult	0.20	0.18	**0.10**	50%	Motion blur
	mean	0.14	0.13	**0.09**	36%	-
TUM-VI	Outdoors2	133.46	106.60	**100.26**	25%	Multi-dynamic objects, Violent rotation at the start and end
	Outdoors3	36.99	30.18	**26.51**	28%	Dim scene, Barren land and forest, Violent rotation at the start and end
	Outdoors6	133.60	99.44	**91.85**	31%	Dim scene, Barren land and forest, Violent rotation at the start and end
	Magistrale1	2.19	1.89	**1.65**	25%	Frequent illumination changes, Outside hallway (weak texture), Violent rotation at the start and end
	Magistrale4	5.12	2.69	**1.28**	75%	Frequent illumination changes, Few dynamic objects, Barren land and forest, Violent rotation at the start and end
	Magistrale5	0.85	1.36	**0.61**	28%	Dim scene, Few dynamic objects, Violent rotation at the start and end
	mean	52.04	40.36	**37.03**	29%	-

* The best result or the smallest error is **bold**.
* Improvement = (RMSE of VINS-Mono – RMSE of OURS) / RMSE of VINS-Mono.

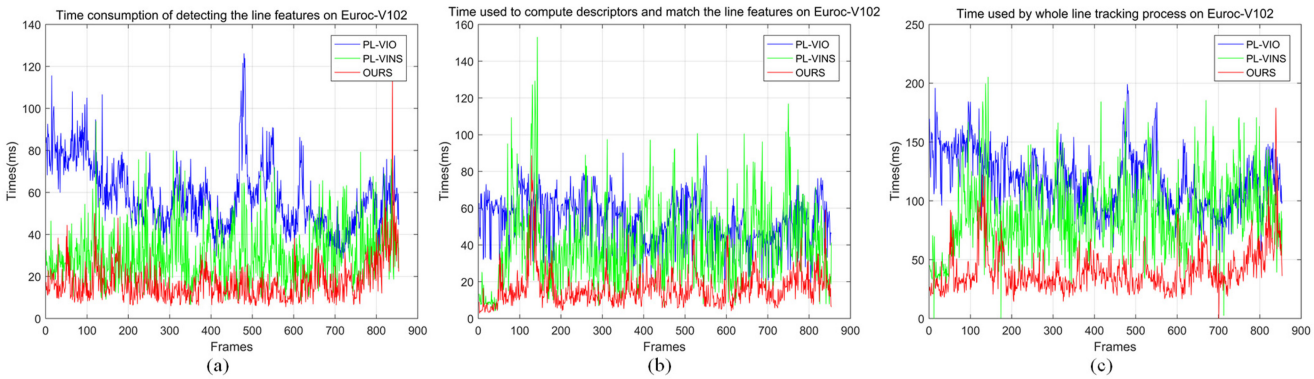


FIGURE 8. Running times of OURS, PL-VIO and PL-VINS on each part of line feature tracking process tested on V1_02_medium.

TABLE 3. Average time consumption (ms) per frame of VINS-Mono, PL-VIO, PL-VINS and the proposed system on V1_02_medium of the EuRoC MAV dataset.

	VINS-Mono	PL-VIO	PL-VINS	OURS
Whole Point Tracking Process	15	15	15	15
Line Detection	×	57	31	17
Line Description and Line Matching	×	53	40	17
Whole Line Tracking Process	×	115	89	41
Local VIO	42	48	46	46
Loop Closure	200	×	200	200

* The symbol "×" indicates that the system does not have the process.

line feature extraction method to extract line features and the proposed line feature matching method to track line features.

It is noticed that our system does not need to compute descriptors. We compare the total time for describing and matching, and the time consumption of our VINS system for describing is zero. As shown in Fig. 8, our VINS system extracts line features and tracks line features and the whole line tracking process on each frame in about 20, 20, and 40 ms. PL-VINS and PL-VIO take about two and three times longer, respectively, than our VINS system for these three processes. In terms of the stability of the time consumption on each frame, PL-VIO and PL-VINS have many

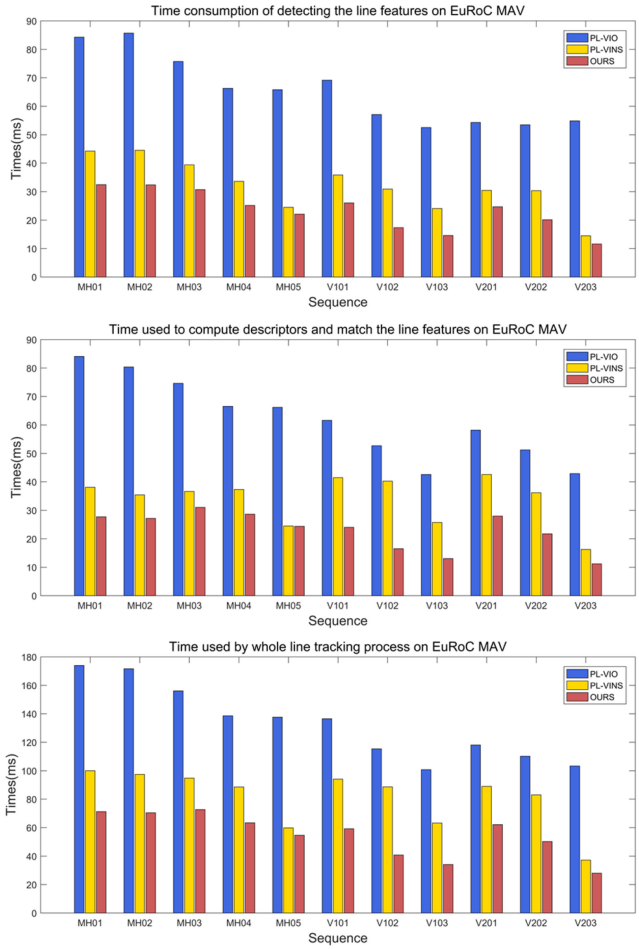


FIGURE 9. Average running time of OURS, PL-VIO and PL-VINS on each part of line feature tracking process tested on all sequences of the EuRoC MAV dataset.

outliers, while our VINS system fluctuates steadily around a certain value. As shown in Fig. 9, in terms of average time consumption per frame on eleven sequences, the highest real-time performance improvement of our VINS system is 0.21, 0.26, 0.27 times that of PL-VIO on V2_03_difficult and 0.56, 0.41, 0.46 times that of PL-VINS on V1_02_medium for line feature extraction, tracking and the whole tracking process, respectively. As shown in the experiments of the real-time performance, our VINS system has the best real-time performance than the other two systems in extracting line features, describing and matching line features and the whole line feature tracking process.

Table 3 provides the time consumption of the three systems for line tracking process and other processes on V1_02_medium of the EuRoC MAV dataset. The three point-line based systems are obtained by introducing line features with the help of VINS-Mono. Except for the line tracking process, the other processes of the four systems in Table 3 are very similar, so their time consumption in these processes are very close. However, the proposed system run more than three times faster to extract and match line features than PL-VIO. And the proposed system is about two times as fast as PL-VINS to extract and match line features.

VI. CONCLUSION

This paper presents a real-time and robust point-line based monocular VINS system. The VINS system is evaluated by on the public datasets of EuRoC MAV and TUM-VI. In the front-end, EDLines with adaptive gamma correction has advantages of extracting a larger ratio of long line features in real time when compared with LSD and FLD. The proposed line feature matching method has fast tracking speed when compared with the KNN method based on LBD descriptors. In the back-end, accurate camera states are obtained by minimizing the cost function containing point-line reprojection errors and IMU residuals. The experimental results show that our VINS system has higher localization accuracy and real-time performance than other advanced systems. In addition, our VINS system shows good robustness in challenging scenes.

REFERENCES

[1] T. Bailey and H. Durrant-Whyte, "Simultaneous localization and mapping (SLAM): Part II," *IEEE Robot. Autom. Mag.*, vol. 13, no. 3, pp. 108–117, Sep. 2006.
[2] M. Wen et al., "Private 5G networks: Concepts, architectures, and research landscape," *IEEE J. Sel. Topics Signal Process.*, vol. 16, no. 1, pp. 7–25, Jan. 2022.
[3] W. Saad, M. Bennis, and M. Chen, "A vision of 6G wireless systems: Applications, trends, technologies, and open research problems," *IEEE Netw.*, vol. 34, no. 3, pp. 134–142, May/Jun. 2020.
[4] B. Huang, J. Zhao, and J. Liu, "A survey of simultaneous localization and mapping with an envision in 6G wireless networks," 2019, *arXiv:1909.05214*.
[5] W. Jiang, B. Han, M. A. Habibi, and H. D. Schotten, "The road towards 6G: A comprehensive survey," *IEEE Open J. Commun. Soc.*, vol. 2, pp. 334–366, 2021.
[6] M. Wen, E. Basar, Q. Li, B. Zheng, and M. Zhang, "Multiple-mode orthogonal frequency division multiplexing with index modulation," *IEEE Trans. Commun.*, vol. 65, no. 9, pp. 3892–3906, Sep. 2017.
[7] J. Li et al., "Monocular 3D object detection based on depth guided local convolution for smart payment in D2D systems," *IEEE Internet Things J.*, early access, Nov. 16, 2021, doi: 10.1109/JIOT.2021.3128440.
[8] G. Bresson, Z. Alsayed, L. Yu, and S. Glaser, "Simultaneous localization and mapping: A survey of current trends in autonomous driving," *IEEE Trans. Intell. Veh.*, vol. 2, no. 3, pp. 194–220, Sep. 2017.
[9] Y. Ling and S. Shen, "Building maps for autonomous navigation using sparse visual SLAM features," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2017, pp. 1374–1381.
[10] C. Liu, W. Wei, B. Liang, X. Liu, W. Shang, and J. Li, "ConvMLP-mixer based real-time stereo matching network towards autonomous driving," *IEEE Trans. Veh. Technol.*, early access, Sep. 14, 2022, doi: 10.1109/TVT.2022.3206612.
[11] Y. Gao et al., "Joint optimization of depth and ego-motion for intelligent autonomous vehicles," *IEEE Trans. Intell. Trans. Syst.*, early access, Mar. 24, 2022, doi: 10.1109/TITS.2022.3159275.
[12] 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, p. 1159, 2018. [Online]. Available: http://www.mdpi.com/1424-8220/18/4/1159
[13] 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.
[14] 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.
[15] Q. Fu et al., "PL-VINS: Real-time monocular visual–inertial SLAM with point and line features," 2020, *arXiv:2009.07462*.
[16] G. Pan, Y. Fan, and Y. Guo, "A low-texture monocular visual odometer based on point-line feature," in *Proc. 5th Int. Conf. Robot. Autom. Sci. (ICRAS)*, 2021, pp. 216–220.

[17] C. Akinlar and C. Topal, "EDLines: A real-time line segment detector with a false detection control," *Pattern Recognit. Lett.*, vol. 32, no. 13, pp. 1633–1642, 2011.
[18] R. G. Von Gioi, J. Jakubowicz, J.-M. Morel, and G. Randall, "LSD: A fast line segment detector with a false detection control," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 4, pp. 722–732, Apr. 2010.
[19] J. H. Lee, S. Lee, G. Zhang, J. Lim, W. K. Chung, and I. H. Suh, "Outdoor place recognition in urban environments using straight lines," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, Jun. 2014, pp. 5550–5557.
[20] J. Y. Bouguet, *Pyramidal Implementation of the Lucas Kanade Feature Tracker Description of the Algorithm*, Intel Corp., Microprocess. Res. Labs, Mountain View, CA, USA, 1999.
[21] M. Burri et al., "The EuRoC micro aerial vehicle datasets," *Int. J. Robot. Res.*, vol. 35, pp. 1157–1163, Jan. 2016.
[22] D. Schubert, T. Goll, N. Demmel, V. Usenko, J. Stückler, and D. Cremers, "The TUM VI benchmark for evaluating visual–inertial odometry," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2018, pp. 1680–1687.
[23] D. G. Lowe, "Object recognition from local scale-invariant features," in *Proc. 7th IEEE Int. Conf. Comput. Vis. (ICCV)*, vol. 2. Kerkyra, Greece, Sep. 1999, pp. 1150–1157.
[24] H. Bay, T. Tuytelaars, and L. V. Gool, "SURF: Speeded up robust features," in *Proc. 9th Eur. Conf. Comput. Vis. (ECCV)*, May 2006, pp. 404–417.
[25] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: An efficient alternative to SIFT or SURF," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Barcelona, Spain, Nov. 2011, pp. 2564–2571.
[26] J. Shi and C. Tomasi, "Good features to track," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 1994, pp. 593–600.
[27] R. Gomez-Ojeda, J. Briales, and J. González-Jiménez, "PL-SVO: Semi-direct monocular visual odometry by combining points and line segments," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2016, pp. 4211–4216.
[28] C. Forster, M. Pizzoli, and D. Scaramuzza, "SVO: Fast semi-direct monocular visual odometry," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, Hong Kong, Jun. 2014, pp. 15–22.
[29] R. Gomez-Ojeda and J. Gonzalez-Jimenez, "Robust stereo visual odometry through a probabilistic combination of points and line segments," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2016, pp. 2521–2526.
[30] R. Gomez-Ojeda, D. Zuñiga-Noël, F.-A. Moreno, 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.
[31] 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 Represent.*, vol. 24, no. 7, pp. 794–805, 2013.
[32] F. Zheng, G. Tsai, Z. Zhang, S. Liu, C.-C. Chu, and H. Hu, "Trifo-VIO: Robust and efficient stereo visual inertial odometry using points and lines," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Madrid, Spain, Oct. 2018, pp. 3686–3693.
[33] A. Kaehler and G. Bradski, *Learning OpenCV*, 2nd ed. Sebastopol, CA, USA: O'Reilly Media, 2014.
[34] S. Lynen, M. W. Achtelik, S. Weiss, M. Chli, and R. Siegwart, "A robust and modular multi-sensor fusion approach applied to MAV navigation," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Nov. 2013, pp. 3923–3929.
[35] P. Corke, J. Lobo, and J. Dias, "An introduction to inertial and visual sensing," *Int. J. Robot. Res.*, vol. 26, no. 6, pp. 519–535, Jun. 2007.
[36] M. Li and A. I. Mourikis, "High-precision, consistent EKF-based visual–inertial odometry," *Int. J. Robot. Res.*, vol. 32, no. 6, pp. 690–711, 2013.
[37] A. Khodadadi, A. Mirabadi, and B. Moshiri, "Assessment of particle filter and Kalman filter for estimating velocity using odometery system," *Sens. Rev.*, vol. 30, no. 3, pp. 204–209, 2010.
[38] M. Bloesch, S. Omari, M. Hutter, and R. Siegwart, "Robust visual inertial odometry using a direct EKF-based approach," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2015, pp. 298–304.
[39] K. Sun et al., "Robust stereo visual inertial odometry for fast autonomous flight," *IEEE Robot. Autom. Lett.*, vol. 3, no. 2, pp. 965–972, Apr. 2018.
[40] A. I. Mourikis and S. I. Roumeliotis, "A multi-state constraint Kalman filter for vision-aided inertial navigation," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, Roma, Italy, Apr. 2007, pp. 3565–3572.
[41] S. Leutenegger, S. Lynen, M. Bosse, R. Siegwart, and P. Furgale, "Keyframe-based visual–inertial odometry using nonlinear optimization," *Int. J. Robot. Res.*, vol. 34, no. 3, pp. 314–334, 2015.
[42] S. Agarwal and K. Mierle. "Ceres solver." 2022. [Online]. Available: http://ceres-solver.org
[43] J. Civera, A. J. Davison, and J. M. M. Montiel, "Inverse depth parametrization for monocular SLAM," *IEEE Trans. Robot.*, vol. 24, no. 5, pp. 932–945, Oct. 2008.
[44] A. Bartoli and P. Sturm, "Structure-from-motion using lines: Representation, triangulation, and bundle adjustment," *Comput. Vis. Image Understand.*, vol. 100, no. 3, pp. 416–441, 2005.
[45] V. Lepetit, F. Moreno-Noguer, and P. Fua. "EPnP: An accurate O(n) solution to the PnP problem," *Int. J. Comput. Vis.*, vol. 81, no. 2, pp. 155–166, Feb. 2009.
[46] G. Sibley, L. Matthies, and G. Sukhatme, "Sliding window filter with application to planetary landing," *J. Field Robot.*, vol. 27, no. 5, pp. 587–608, Sep. 2010.
[47] M. Calonder, V. Lepetit, C. Strecha, and P. Fua, "BRIEF: Binary robust independent elementary features," in *Proc. Eur. Conf. Comput. Vis.*, Crete, Greece, Sep. 2010, pp. 778–792.
[48] E. Rosten, R. Porter, and T. Drummond, "Faster and better: A machine learning approach to corner detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 1, pp. 105–119, Jan. 2010.
[49] D. Galvez-López and J. D. Tardos, "Bags of binary words for fast place recognition in image sequences," *IEEE Trans. Robot.*, vol. 28, no. 5, pp. 1188–1197, Oct. 2012.
[50] P. Babakhani and P. Zarei, "Automatic gamma correction based on average of brightness," *Adv. Comput.*, vol. 4, no. 6, pp. 156–159, 2015.
[51] M. Andrew, "Multiple view geometry in computer vision," *Kybernetes*, vol. 30, nos. 9–10, pp. 1865–1872, 2001.
[52] P. J. Huber, "Robust estimation of a location parameter," *Ann. Math. Stat.*, vol. 35, no. 2, pp. 73–101, 1964.

ZHENFEI KUANG received the bachelor's degree in measurement and control technology and instrumentation from the Nanchang Institute of Technology in 2020. He is currently pursuing the master's degree with the School of Electronics and Communication Engineering, Guangzhou University. His current research interest includes visual inertial simultaneous localization and mapping.

WEI WEI (Member, IEEE) received the B.S. degree in optical information science and technology from Jilin University in 2010, and the Ph.D. degree in electronic science and technology from the Beijing University of Posts and Telecommunications in 2015. He worked as a Postdoctoral Researcher with University College Cork and Tyndall National Institute from 2015 to 2016, a Research Engineer with Huawei from 2016 to 2017, and a Hong Kong Scholar Researcher with The Hong Kong Polytechnic University from 2018 to 2020. He is currently an Associate Professor with the School of Electronics and Communication Engineering, Guangzhou University. His current research interests include LiDAR, depth estimation, and autonomous driving.

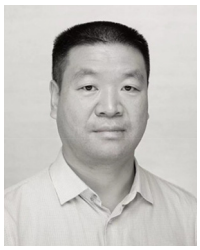
YIER YAN received the B.S. degree in applied electronics from South Central Nationality University, Wuhan, Hubei, China, and the M.S. and Ph.D. degrees in communication engineering from Chonbuk National University, Jeonju, South Korea. He is currently working with the School of Mechanical and Electrical Engineering, Guangzhou University, China. His research interests include information theory, signal processing, and OFDM in MIMO system.

JIE LI received the bachelor's degree in electronic information engineering from Suzhou University in 2020. He is currently pursuing the master's degree with the School of Electronics and Communication Engineering, Guangzhou University, Guangzhou, China. His current research interest includes the simultaneous localization and mapping based on vision.

GUANGMAN LU received the bachelor's degree in communication engineering from the Guangxi University for Nationalities in 2020. He is currently pursuing the master's degree with the School of Electronics and Communication Engineering, Guangzhou University. His current research interest includes the simultaneous localization and mapping based on 3-D LiDAR.

YUYANG PENG (Senior Member, IEEE) received the M.S. and Ph.D. degrees in electrical and electronic engineering from Chonbuk National University, Jeonju, South Korea, in 2011 and 2014, respectively. He worked as a Postdoctoral Research Fellow with the Korea Advanced Institute of Science and Technology, Daejeon, South Korea, from 2014 to 2018. He is currently an Assistant Professor with the School of Computer Science and Engineering, Macau University of Science and Technology, Macau, China. His research activities lie in the broad area of digital communications, wireless sensor networks, and computing. In particular, his current research interests include cooperative communications, spatial modulation, and energy optimization. He served as an Editor for IEEE ACCESS.

JUN LI (Member, IEEE) received the B.S. degree from the South Central University for Nationalities, Wuhan, China, in 2009, and the Ph.D. degree from Chonbuk National University, Jeonju, South Korea, in 2016. He is currently an Associate Professor with the School of Electronics and Communication Engineering, Guangzhou University, Guangzhou, China. He has published more than 60 papers in refereed journals and conference proceedings. His research interests include spatial modulation and OFDM with index modulation.



WENLI SHANG (Member, IEEE) was born in Heilongjiang, China, in 1974. He received the M.S. degree from the School of Mechanical and Automation Engineering, Northeastern University in 2002, and the Ph.D. degree from the Laboratory of Industrial Control Network and System, Shenyang Institute of Automation, Chinese Academy of Sciences in 2005. From 2005 to 2019, he served as an Assistant Researcher, an Associate Researcher, and a Researcher with the Shenyang Institute of Automation, Chinese Academy of Sciences. Since 2020, he has been a Professor with Guangzhou University. His research interests include computational intelligence and machine learning, edge computing, and industrial control system information security.