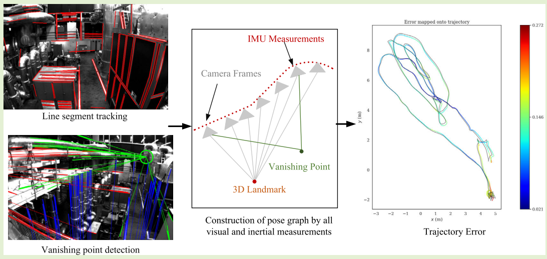
IEEE Sensors Council

HTSR-VIO: Real-Time Line-Based Visual-Inertial Odometry With Point-Line Hybrid Tracking and Structural Regularity

Bing Zhang, *Graduate Student Member, IEEE*, Yankun Wang, *Graduate Student Member, IEEE*, Weiran Yao, *Member, IEEE*, and Guanghui Sun, *Senior Member, IEEE*

***Abstract*—Line features in artificial scenes encode geometric information, such as parallelism and orthogonality that can provide excellent observational landmarks for localization and navigation. However, line features extracted in challenging scenes face problems contain duplicate detection and incomplete structural line segment constraints. To address these problems, we propose an efficient visual-inertial odometry with point-line hybrid tracking and structural regularity. First, a novel point-line hybrid tracking algorithm is given, which utilizes matching relations between the current and previous moments of tracked points to support nearest-neighbor line segment tracking. Then, a line-point binding (LPB) residual error that is positively correlated with point feature reprojection error is proposed, and we incorporate this error into the sliding window optimization for pose estimation. Besides, to fully exploiting structural regularity of line features, we extract vanishing points in multiple views to separate vertical lines from spatial structured lines. Spatial consistency constraints between these vertical lines and gravity vector measured by the inertial measurement unit (IMU) are utilized to refine the body pose. Experimental results with the state-of-the-art (SOTA) methods indicate that the proposed method can improve the accuracy of monocular visual-inertial odometry to at least 13.1%.**



***Index Terms*— Line tracking, point-line fusion, vanishing point, visual-inertial odometry.**

I. Introduction

ESTIMATING six-DoF pose of agent robustly and accurately is a fundamental and challenging problem in navigation and localization [1], [2], [3]. Recently, researchers have been attracted by visual simultaneous localization and mapping (VSLAM) and visual inertial odometry (VIO), which can provide relatively high-precision motion estimation by capturing camera images [4], [5]. Further, there are various ways to improve the performance of visual localization [6], such as employing novel visual sensors like event camera [7] and omnidirectional cameras to obtain more information, using feature reprojection-based methods [8] or photometric error-based methods [9] for pose estimation. In this article, we focus on feature-based monocular visual localization that can be generalized to pinhole and fisheye cameras. In VSLAM/VIO, the feature-based method, particularly point feature [10], plays a crucial role in camera motion estimation since it is computationally efficient and widely available [11], [12]. However, it has been confirmed that point features faced tracking failure and weak geometric limitations in sparse texture environment.

To overcome the weakness of point features, some researchers have suggested incorporating line features into SLAM since texture loss often occurs in artificial environments [13], [14], [15]. Line features [16] can be leveraged as a supplement with structural information and establish the line feature observation error model to optimize body pose [17], [18], [19]. However, some line features have significant drawbacks such as incomplete detection, partial occlusion, and insufficient constraints, which may lead to incorrect observation models. Pure line feature methods are even more unstable than point feature methods under noise interference.

To address these challenges, some advanced algorithms for detecting and tracking line features are proposed to make

Manuscript received 19 September 2023; accepted 20 February 2024. Date of publication 29 February 2024; date of current version 2 April 2024. This work was supported in part by the National Natural Science Foundation of China under Grant U23A20346, Grant 62033005, Grant 62173107, and Grant 62106062; and in part by the Key Project of Natural Science Foundation of Heilongjiang Province under Grant ZD2021F001. The associate editor coordinating the review of this article and approving it for publication was Dr. Baoding Zhou. *(Corresponding author: Guanghui Sun.)*

The authors are with the School of Astronautics, Harbin Institute of Technology, Harbin, Heilongjiang 150001, China (e-mail: 20b904048@stu.hit.edu.cn; 20b904057@stu.hit.edu.cn; yaoweiran@hit.edu.cn; guanghuisun@hit.edu.cn).

Digital Object Identifier 10.1109/JSEN.2024.3369095

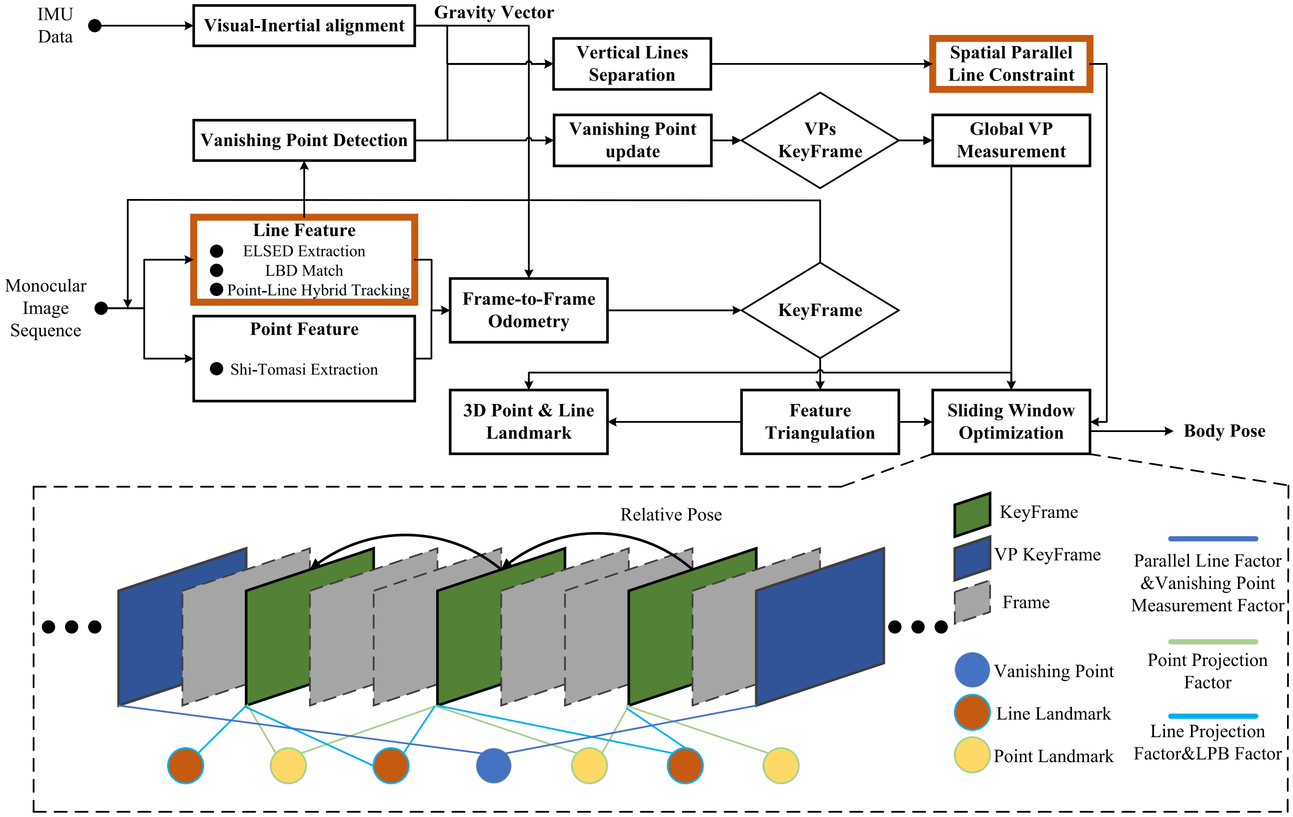


Fig. 1. Pipeline of our proposed system.

line feature-based SLAM more reliable and precise. Zhao and Vela [20] apply a good line cutting method to adjust the weight of line features and estimate the camera pose well even when the texture is poor or the image is blurry. Line segment flow [21] is proposed to model how line segments change over space and time, which helps to track line features by following their spatiotemporal patterns. IMU-KLT line prediction algorithm [22] which uses the optical flow method and inertial measurement unit (IMU) preintegration, is applied for feature prediction matching with good reliability. EPLF-VINS [23] proposed an efficient line optical flow feature tracker based on the grayscale invariance assumption, which guaranteed quality of line features and speed of line tracking.

In addition, some other researchers have concentrated on incorporating the structural constraints of line features into SLAM/VIO systems in order to enhance the performance of line feature-based methods. As a result, vanishing point detection algorithms are utilized in SLAM to extract structural lines from images [24], [25]. In the perspective projection model, spatially parallel 3-D lines are projected to the image plane onto intersect with a point, which is called the vanishing point. The Manhattan hypothesis states that there exist vanishing points in a global frame with three mutually orthogonal dominant directions, which is consistent with most man-made environments. Using vanishing points, the image lines can be classified into three mutually orthogonal groups. In the field of SLAM, VP-pass [26] uses vanishing points as global virtual landmarks in indoor buildings to correct the local observation error. Structure-VIO [27] is proposed to make full use of structural lines in an artificial environment. This system is based on an extended Kalman filter, which effectively reduces the orientation error. A two-parameter method is proposed to represent structured and unstructured lines [28], so as to reduce the computational effort of triangulation and pose optimization. UV-SLAM [29] proposed a structural line system without the constraints of the Manhattan assumption to extend the application scenario of vanishing points in SLAM.

However, current line feature-based methods are still far from perfect. First, few studies have looked at how to track both point and line features together and make them better in point-line fusion SLAM. Most studies measure the errors of point and line features separately and do not link them in the same image when they optimize the sliding window. Second, the vanishing point measurement model is only related to the orientation part of transformation, which cannot fully utilize the structural information of spatial lines. Motivated by the discussion above, we propose a real-time visual-inertial odometry with point-line hybrid tracking and structural regularity, as shown in Fig. 1, referred to as HTSR-VIO.

Compared with the previous line-based SLAM works, our method employs matched points to predict line segments and increases the number of matched image lines using neighbor line matching and LBD matching algorithms in parallel. Relying on this tracking approach, a line-point binding (LPB) residual error using nearby points of lines is proposed to refine body pose. In addition, we propose a novel vanishing point update mechanism to estimate vanishing points as variables and construct a spatial geometric consistency function for structural lines using the property that vertical structural lines are colinear with the gravity direction in a structured environment, thus optimizing the transformation matrix instead of the rotation matrix alone. To validate our proposed method, we use EuRoC public dataset [30], Kaist-VIO dataset [31] and NTU-VIRAL dataset [32] to test the proposed method and compare with VINS-mono, PL-VINS, and UV-SLAM. The main contributions of HTSR-VIO are as follows.

1) We propose a novel visual-inertial estimator framework that utilizes point-line hybrid tracking patterns and vanishing point constraints to achieve accurate state estimation.
2) A point-line hybrid tracking algorithm is proposed to track line segments by obtaining the matching relationship of neighbor points. Then, a new LPB residual error is proposed for neighbor points of the line feature to expand the observation model. Using the proposed method, long-term tracking line features can be obtained to improve the localization accuracy of this system.
3) An efficient vanishing point update mechanism with vanishing point keyframe strategy is designed to maintain the accuracy of global observation estimation, which can cope with the situation that a vanishing point is lost due to long-term unobservation.
4) An observation residual error model based on the vanishing point measurement is constructed to reduce rotation drift. To fix the position offset, we use a set of vertical spatial lines and a gravity vector measured by IMU to establish the spatial geometric consistency loss function, which helps us refine the pose estimation.

The rest of this article is organized as follows. We present an overview of the proposed system in Section II. The details of the proposed point-line hybrid tracking algorithm and LPB residual error are illustrated in Section III. In Section IV, we elaborate the vanishing points update mechanism, vanishing point observation residual error model, and spatial geometric consistency of vertical lines. Experimental results on public dataset are provided in Section V. Conclusions and future work are given in Section VI.

II. System Overview

A. Framework of System

Building upon PL-VINS as our foundation, we present a novel system that integrates point features, unstructured line features, and structural line features to enhance the performance of VIO. We take IMU measurements and visual images as inputs, visual observations and IMU measurements are aligned to complete the initialization. Then, on the one hand, the feature reprojection error is established by tracking line features and point features. In this process, we propose a point-line hybrid tracking algorithm which predicts line segments based on the matching relationships of neighboring point features and performs neighbor matching on line features. The algorithm is operated in parallel with the LBD matching algorithm, combining their matching results to boost the number of matching lines. Meanwhile, leveraging the positional relationships between tracked points and lines, we establish a LPB residual error which connects the point reprojection error model and the line observation model, thereby reinforcing constraints in the vicinity of image line segments. On the other hand, the structure lines are extracted by the vanishing point detection method which is improved by the robust discriminant module, and the vanishing point keyframe strategy is designed to update the global vanishing point. Furthermore, the gravity direction measured by IMU is used to separate the structure lines in the vertical direction for developing the spatial consistency, which is added into the sliding window to optimize the keyframe pose together with the residual error of the vanishing point observation.

B. Problem Statement

The problem we need to solve is how to robustly implement line segment tracking and how to effectively integrate vanishing point observation into the SLAM system to obtain accurate poses.

A spatial line is parameterized using a 3-D infinite line in plücker coordinate and optimized by an orthogonal representation. During feature tracking, the projection between the 3-D spatial lines and the observed 2-D line segments is related to the camera pose and landmark observations. While the initialization of the system is completed, we select accurate vanishing points for initialization and make them as global variables. In the subsequent vanishing point tracking process, we update the global vanishing points by adjusting the rotation matrix of pose. In practice, the projection of spatial lines may differ from the corresponding image line segments observed due to occlusion and blurred appearance. Because of this deficiency, we design the point-line hybrid tracking module (in Section III-B) to solve these problems. In addition, the vanishing point has the intrinsic property that measurement error is related only to the rotational matrix. To extend the functionality of vanishing point measurement in SLAM systems, we use the gravity vector measured by IMU to construct spatial consistency constraints to refine the camera poses.

In this work, given the available visual and inertial measurements $\mathcal{Z}_n$ and priors $\mathcal{X}_0$, the posterior probability of state variables $\mathcal{X}_n$ can be represented by the probabilistic model of Bayesian networks as follows:

$$\begin{aligned} P\left(\mathcal{X}_n|\mathcal{Z}_n\right) &\propto P\left(\mathcal{X}_0\right) P\left(\mathcal{Z}_n|\mathcal{X}_n\right) \\ &= P\left(\mathcal{X}_0\right) \prod_{(i,j,k)\in\mathcal{N}_n} P\left(\mathcal{C}_i, \mathcal{L}_i, \mathcal{I}_{ij}, \mathcal{V}_k|\mathcal{X}_n\right) \\ &= P\left(\mathcal{X}_0\right) \prod_{(i,j)\in\mathcal{N}_n} P\left(\mathcal{I}_{ij}|x_i, x_j\right) \prod_{i\in\mathcal{N}_n} \prod_{c\in\mathcal{C}_i} P\left(z_{ic}|x_i\right) \\ &\quad \times \prod_{i\in\mathcal{N}_n} \prod_{l\in\mathcal{L}_i} P\left(z_{il}|x_i\right) \prod_{k\in\mathcal{N}_n} \prod_{v\leq 3} P\left(z_{kv}|x_k\right) \end{aligned} \tag{1}$$

where $\mathcal{N}_n$ indicates the set of all keyframes up to time n, $\mathcal{X}_n$ is the state of all keyframes. x_i is the state of keyframe at time i. $\{z_{ic} \in \mathcal{C}_i\}$, $\{z_{il} \in \mathcal{L}_i\}$ denote the point feature measurement and the line feature measurement, respectively, at keyframe i. $\{z_{kv} \in \mathcal{V}_k\}$ is the vanishing point measurement at vanishing point keyframe k.

Then, (1) is transformed to a negative log-posterior function to solve $\hat{\mathcal{X}}_n$ under the assumption of zero-mean Gaussian noise. The negative log-posterior can be written as a sum of squared residual errors to estimate the six-DoF state

$$\hat{\mathcal{X}}_n = \arg\min_{\mathcal{X}_n} \left\{ \|r_0(\mathcal{X})\|^2 + \sum_{(i,j)\in\mathcal{N}_n} \left\| r_I\left(\hat{\mathcal{I}}_{ij}, \mathcal{X}\right)\right\|^2 \right. + \sum_{i\in\mathcal{N}_n}\sum_{c\in\mathcal{C}_i} \|r_P(\hat{z}_{ic}, \mathcal{X})\|^2 + \sum_{i\in\mathcal{N}_n}\sum_{l\in\mathcal{L}_i} \|r_L(\hat{z}_{il}, \mathcal{X})\|^2_{P^{c_i}_{l_j}} + \sum_{k\in\mathcal{N}_n}\sum_{v\le 3}\left\|r_V\left(\hat{z}_{kv}, \hat{z}^g_{kv}, \mathcal{X}\right)\right\|^2 + \sum_{k\in\mathcal{N}_n}\sum_{s\in\mathcal{S}_k}\left\|r_S\left(\hat{z}_{ks}, \mathcal{X}\right)\right\|^2 \left. + \sum_{i\in\mathcal{N}_n}\sum_{l\in\mathcal{L}_i}\sum_{c\in\mathcal{C}_i}\left\|r_B\left(\hat{z}_{il}, \hat{z}_{ic}, \mathcal{X}\right)\right\|^2 \right\} \quad (2)$$

where $\mathcal{X}$ represents state that need to be solved. r_0, r_I, r_P, r_L, and r_V are the residual errors of the corresponding measurement. r_B and r_S are the proposed residual errors that associated with the LPB and structural consistency, respectively. The purpose of Sections III-C and IV-C is to provide expressions for r_B, r_S. $\{\hat{z}_{ks} \in \mathcal{S}_k\}$ denotes the vertical line measurement at vanishing point keyframe k. $\hat{z}^g_{kv}$ is the global vanishing point measurement that updates constantly as keyframes are added.

III. Proposed Point-Line Hybrid Tracking Algorithm

A. Point-Line Feature Extraction

In the image, point features are usually extracted from structural edges or corners. We use the Shi–Tomasi algorithm [33] to quickly detect corner points with sharp changes in edge grayness.

Line features are detected at the edges of structures in images, which are usually used for structure detection in artificial environments. We use the enhanced line segment drawing algorithm (ELSED) [34], an efficient local line segment detector available, which uses a local segmental growth algorithm to connect gradients to align the discontinuities between pixels. Since its initial goal is structure detection, we have improved the original ELSED algorithm to make it suitable for SLAM. Specially, we adjust the parameter selection of the discontinuous in the case of the ELSED algorithm (e.g., Gaussian kernel size adjustment) to ensure that the detected line features are reliable. Compared with the previous algorithms, our algorithms are more concerned with the neatness of detected line segments, which helps to improve the performance of our algorithm in reconstructing line maps and detecting vanishing points. To solve the problem of cluttered lines caused by some edges being detected multiple times, we employ near line merging and broken line splicing to enhance the neatness of image line segments. The near line merging strategy refers to merging two line segments that are very closely located in detected lines. The broken line splicing strategy refers to reconstructing two line segments with approximate slopes and similar endpoints into a single line segment.

B. Hybrid Tracking Algorithm

The extracted line features are paired with the line features from the previous frame through the use of the LBD algorithm [35] in order to obtain rough results. Beyond the rough match results, there are some indistinguishable line segments that failed to be matched. Therefore, we propose a matching strategy that utilizes the matching points around the target line segment being tracked to support the matching of line segments. Point features around a line feature in the artificial environment have similar geometric characteristics to this line feature. Matched point features in the surrounding area can be utilized to provide an initial guess for line feature tracking.

Further, the velocity and pixel shift of point features are introduced into prediction and matching of line features. In a very short time (a few milliseconds), the camera motion can be regarded as a uniform motion. Therefore, point features can be considered to move with a constant velocity on the image plane during this short time. Based on this assumption, we compute the velocities of point features on the image plane to predict their neighbor line features. We first set a rectangular mask centered on the target line feature and search for point features in this region. If there are enough points located in this region, we take points in this region and their corresponding points from previous frame as sets $\mathbb{Q}_1$ and $\mathbb{Q}_2$. R and t describe relative rotation and translation between them. K is the camera projection matrix

$$s_2 P_2 = s_1(R P_1 + t). \quad (3)$$

We transform $\mathbb{Q}_1$ and $\mathbb{Q}_2$ by K^{-1} to obtain normalized coordinates. Then, we obtain

$$s_2 K^{-1}\mathbb{Q}_2 = s_1(R K^{-1}\mathbb{Q}_1 + t). \quad (4)$$

In (4),we denote these two points set depth as s_1 and s_2. For two consecutive frames in a very short time, we consider $s_1 = s_2$. Therefore, (4) can be simplified as

$$\mathbb{Q}_2{}^T K^{-T} t^\wedge R K^{-1}\mathbb{Q}_1 = 0. \quad (5)$$

We use SVD method to solve the optimal 3-D rotation matrix R in the above problem, and then solve the translation vector t by R. Relying on rotation matrix and translation vector of the rectangular mask, $l_i \in \mathbb{C}_1$ is predicted roughly to be l'_i in $\mathbb{C}_2$. Then, the prediction of line segment is adjusted by regional pixel velocity $V_{i\to j}$ includes orientation and magnitude that are determined by point features located in this region

$$\left|1 - V_{l_i \to l'_i} \big/ V_{i\to j}\right| \le \tau. \quad (6)$$

After short-term prediction of line features, we detail the neighbor line matching. At first, a search region is established with the midpoint of the predicted line segment as the center. By comparing the endpoint distance, midpoint distance, and angle error between the predicted line segment and the line segment to be matched, we select the line segment with higher

near line merging and broken line splicing

利用预积
分计算的V
在预积分
辅助当中
和Rt做一
个加权或
者Rt丢失
的时候或
者做一个
滤除？

Algorithm 1 Hybrid Tracking Algorithm With Short-Term Prediction and Neighbor Line Matching

```
**Requirement:** $(I_i, I_j, \mathbb{C}_1, \mathbb{C}_2, \mathbb{Q}_1, \mathbb{Q}_2, K, d_e, d_m, \theta_l)$
**Ensure:** Match result $X_{i\to j}$
    $p_i \in \mathbb{Q}_1, p_j \in \mathbb{Q}_2, l_i \in \mathbb{C}_1, l_j \in \mathbb{C}_2$
    $PointMatchSuccess\,(\mathbb{Q}_1, \mathbb{Q}_2)$
    $\Omega_1 \leftarrow LineMatchFail\left(\mathbb{C}_1, \mathbb{C}_2, LBD_{i\to j}, K\right)$
    $\Omega_2 \leftarrow LineMatchSucces\left(\mathbb{C}_1, \mathbb{C}_2, LBD_{i\to j}, K\right)$
**Step1:** Short-term Line Prediction by Matched Points
    **for** $l_i \in \Omega_1$ do
        $\mathrm{H} \leftarrow RectMask\,(l_i)$
        $(R, t) \leftarrow SVD\,(\mathbb{Q}_1, \mathbb{Q}_2)$
        **for** $p_i \in \mathrm{H}$ do
            $v^p_{i\to j} \leftarrow OpticalFlowVelocity\left(p_i, p_j\right)$
        **end for**
        $v^H_{i\to j} \leftarrow RegionVelocity\left(v^p_{i\to j}\right)$
        $l'_i \leftarrow LinePredict\left(R, t, l_i, v^H_{i\to j}\right)$
        $\mathbb{Z} \leftarrow {l'}_i$
    **end for**
    **return** $\mathbb{Z}$
**Step2:** Neighbor Line Matching Algorithm
    $(l'_i, l_j) \leftarrow ForwardLineMatch\,(\mathbb{Z}, d_e, d_m, \theta_l, \mathbb{C}_2)$
    **for** $l_j \in \mathbb{C}_2$ do
        $\Omega_3 \leftarrow ReverseCheck\left(l_j, \mathbb{Z}, d_m, \theta_l, Score\right)$
    $X_{i\to j} \leftarrow \Omega_2 \cup \Omega_3$
    **return** $X_{i\to j}$
```

matching score as the final matched line feature. Finally, we judge whether the line feature segment predicted by correspondence through reverse verification and regional gray check. The process runs in parallel with the LBD line segment matching algorithm. Then, the LBD matching results and the proposed tracing results are merged. The fusion result of the same matching line segment is determined by the matching score of the hybrid tracking algorithm. If the matching score is higher than a certain threshold, the result of the hybrid tracking matching is considered more reliable; otherwise, the matching line segment of the LBD algorithm is selected as the final result. Thus, more reliable line segments which have long track length can be matched. The whole algorithm is shown in algorithm (1).

C. Pose Estimation Refined by LPB

While the line feature is continuously tracked with the surrounding point features based on a point-line hybrid tracking algorithm, we assume that the line segment feature has a positional relationship with the surrounding point features. Then, the line feature measurement model is rewritten to

$$P\left(\mathbb{L}\right) \propto P\left(T_0\right) \prod_{i,j,k} P\left(T_{i+1} | T_0, \ldots, T_i\right) \times P\left(l^{o,j}_{i+1} | T_{i+1}, L^j, l^j_i, p^k_i, p^{o,k}_{i+1}, v^k_{i+1}\right) \tag{7}$$

where $\mathbb{L}$ indicates variables of line feature observation model, including the current camera pose T_i, the jth 3-D spatial lines L^j, and the corresponding 2-D line feature observations $l^{o,j}$. $p^{o,k}$ denotes the kth point observation in rectangular search region centered target line. v^k_{i+1} is the pixel velocity of point feature $p^{o,k}_{i+1}$ on camera frames from i to $i+1$.

As shown in Fig. 2, the projection of L observed at frame i is $l^{o,j}_i$, and $p^{o,k}_i$ that is near $l^{o,j}_i$. We consider a line feature and surrounding point features to constitute a LPB feature if the projections of L and P observed at frame i are still close enough that the distance between them is less than a threshold in the process of line feature tracking. The points bound to the line segment are not on the line feature, but are similar to the endpoints of the line segment, with the property of dramatic grayscale changes and very close to the line segment. In addition, we remain concerned about the tracking length of LPB features, which refers to the times of consecutive tracking of the same feature. Then, the LPB residual error model r_B is proposed to represent the distance d_{LPB} between line feature and the binding points. If the tracking length of the LPB feature is greater than constant threshold μ_1, the LPB residual error model is adopted to optimize pose. We set the constant threshold μ_1 to five times. In general, the point feature has measurement error in different camera frames to establish the concept of reprojection error. In the LPB minimization model, d_{LPB} and the reprojection error of point are positively correlated. In this article, the consecutive tracking time of the same feature is set as tracking length of this feature, and if the feature tracking length is greater than or equal to 5, we consider this feature as a long-term tracking feature. Through the minimization of LPB residual error, the estimation of long-term landmark and camera posed can be refined

$$\hat{p}^{o,k}_{i+1} = \pi\left(R^c_b R^{b_j}_{b_i} R^b_c P^k + R^c_b R^{b_j}_{b_i} t^b_c + R^c_b R^{b_j}_w \Delta t^w - R^c_b t^b_c\right) \tag{8}$$

$$r_B\left(l^{o,j}_i, l^{o,j}_{i+1}, p^{o,k}_i, p^{o,k}_{i+1}, \mathrm{T_{i+1}}\right) = \sum_k \frac{\left(\left(\hat{p}^{o,k}_{i+1}\right)^T \cdot l^{o,j}_{i+1}\right)^2}{\left\| l^{o,j}_{i+1} \right\|_2}. \tag{9}$$

IV. Structural Line Cluster and Residual

Relying on the long-term robustness of line segment features preserved by hybrid tracking algorithm, we introduce vanishing point detector to classify the image lines and further refine the pose estimation based on the geometric properties of vanishing points.

In Manhattan world, given three spatial parallel lines sets, the direction of these lines must be aligned with the dominant directions of Manhattan world, respectively. We generally consider the 3-D space constituted by the first detected vanishing point as the Manhattan world, denoted as $\mathbb{D} = \{d_k\}^3_{k=1}$. If we know camera projection matrix K, then the relationship between dominant direction of Manhattan world d_k and vanishing point v^i_k at camera frame i can be indicated as

$$v^i_k \propto K R_i d_k. \tag{10}$$

In (10), we obtain two geometric properties that will be used later in the vanishing point update and vanishing point measurement model.

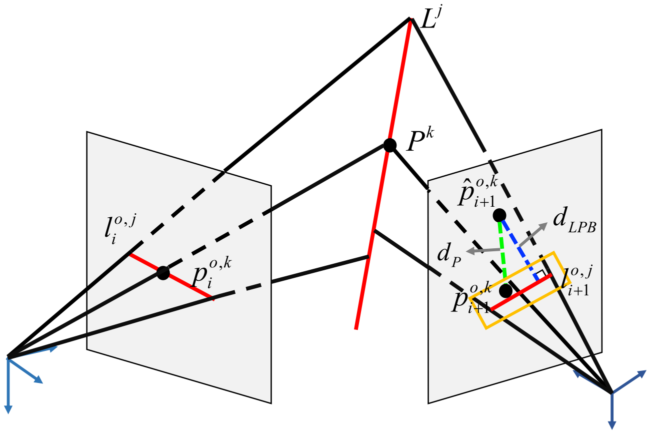


Fig. 2. Illustration of LPB residual error. The yellow box indicates the search range of neighboring points of the LPB feature. The green dotted line in the right image plane indicates the reprojection error of the point feature, blue dotted line represents the distance between the observed line segment $l_{i+1}^{o,j}$, and the binding point $\hat{p}_{i+1}^{o,k}$ at frame $i+1$. The blue dotted line distance is positively correlated with the reprojection error of the point feature.

1) In the Manhattan world, the transformation of the vanishing point between global coordinate and camera coordinate is independent of translation and is only related to the rotation matrix.
2) The 3-D line between the optical center of the camera and the vanishing point is parallel to the spatial lines corresponding to this vanishing point as shown in Fig. 3.

A. Vanishing Point Update

In this article, we focus on the update mechanism of vanishing points and establish a keyframe strategy to optimize the updated vanishing points. Analogous to the application of feature points in SLAM, we divide the update process of vanishing points into: *vanishing point detection*, *vanishing point initialization*, and *vanishing point tracking*.

1) Vanishing Point Detection: We extract vanishing points based on [36]. This algorithm combines data sampling and parameter search method to detect vanishing points online with high accuracy and efficiency. We adopt the pattern that the two image lines sampled belong to the same vanishing point in this method to fix two degrees of freedom, and then use the branch and bound (BnB) algorithm to search for the remaining one degree of freedom. Taking image lines obtained by the ELSED line segment detector as input, three vanishing points are detected, as shown in Fig. 4. 检测到消失点然后检测线段

However, the general vanishing point detection approach need to traverse all image lines with high computational cost. The real line segments in a scene are usually longer (because the structural lines of an interior environment are generally straight), while those extracted from nonreal straight lines should theoretically be shorter due to boundary curvature. Therefore, we set a line segment length threshold to filter them out

$$\text{len} \geq (\text{len}_{\min} = w \cdot \min(W_I, H_I)) \tag{11}$$

where len is the length of segment, W_I and H_I indicate the size of image, and w is scale factor.

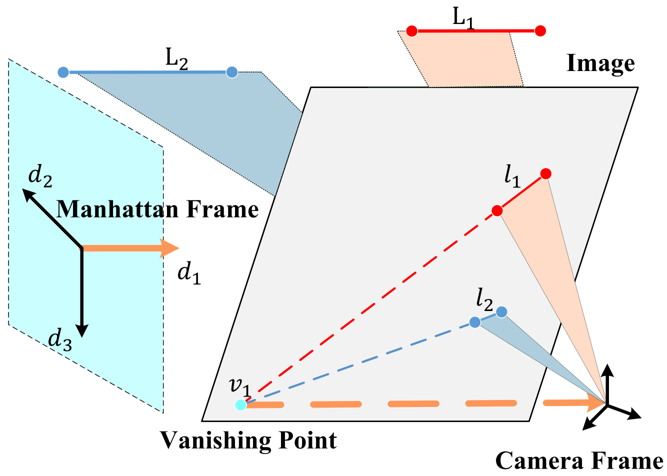
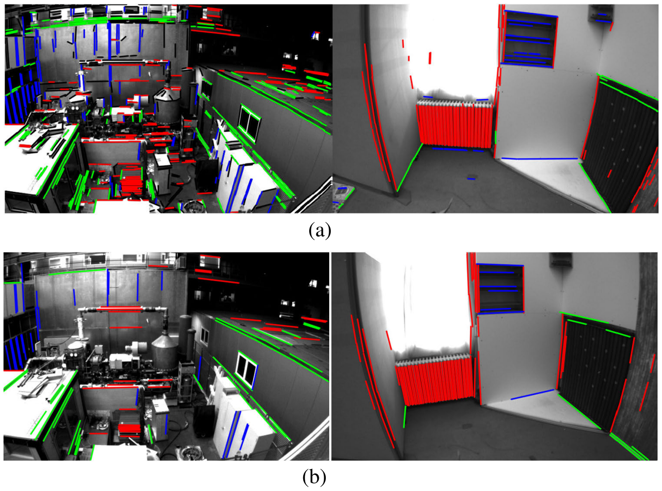


Fig. 3. Illustration of structural constraint of parallelism. v_1 is the vanishing point corresponding to the dominant direction d_1 of Manhattan world.


(a)

(b)

Fig. 4. Comparisons of our origin vanishing point algorithm and improved vanishing point algorithm. In this figure, blue lines, red lines, and green lines represent corresponding lines group clustered by vanishing points. (a) Results of origin vanishing point algorithm. (b) Results of image line segment classification after removing cluttered segments.

Moreover, due to the possibility of line segment clutter in indoor scenes, we construct a vanishing point robust discriminant module to ensure the accuracy of vanishing points. We assume that the total length of line segments extracted from the image is $L_{\text{total}}^{\text{len}}$, and the total length of line segments corresponding to the vanishing points is $L_{\text{vp}}^{\text{len}}$. The number of total line segments and the number of line segments corresponding to vanishing points are $L_{\text{total}}^{\text{num}}$ and $L_{\text{vp}}^{\text{num}}$, respectively. We consider that the image lines corresponding to the vanishing point is neater, and the vanishing point detection is more robust

$$\frac{L_{\text{vp}}^{\text{len}}}{L_{\text{total}}^{\text{len}}} > \alpha, \quad \frac{L_{\text{vp}}^{\text{num}}}{L_{\text{total}}^{\text{num}}} > \beta. \tag{12}$$

2) Vanishing Point Initialization: Based on the current body pose obtained from the frame-to-frame odometry, we perform the initialization of the vanishing point in global coordinate. Note that if less than two vanishing points are extracted, we discard this frame and operate the next one until a qualified frame is found for initialization. If the vanishing points extracted are robust and the current 3-D vanishing points in the global coordinate have not been initialized, then the 3-D coordinates of vanishing points in the global coordinate are

$$v_i^g = R_g^b R_c^b \cdot v_i^c, \quad i < 3. \tag{13}$$

In the joint initialization of the camera and IMU, the gravity vector G refinement is necessary (in Section IV-C). So, we consider the initialization is success if the vanishing point vector and the gravity vector are almost colinear

$$\arccos\left(v_i^g \otimes G\right) > \rho \tag{14}$$

where ρ is the artificially set angle threshold.

3) Vanishing Point Tracking: We observe vanishing points and track them as system states, which enables multiple vanishing point directions to be used simultaneously in tracking process. This method can cope with the situation that a vanishing point direction is lost due to long-term unobservation, such as the case of high-speed and motion drifts, the vanishing point can be reinitialized at this time. The angle $\delta\theta$ between the current vanishing point and the global vanishing point is denoted as

$$\delta\theta_i = \arctan\left(\frac{\left\|\left(R_g^b R_c^b v_i^c\right) \otimes v_i^g\right\|}{\left(R_g^b R_c^b v_i^c\right) \cdot v_i^g}\right), \quad i < 3. \tag{15}$$

Additionally, we take vanishing point distance δd between v^c and v^g as

$$\delta d_i = \left\|\left(R_g^b R_c^b v_i^c\right) - v_i^g\right\|_2, \quad i < 3. \tag{16}$$

If $\delta\theta_i < \phi$ and $\delta d_i < \xi$, we consider the vanishing point tracking is success.

Then, a vanishing point keyframe strategy is designed to reduce the frequency of vanishing point update. We select vanishing point keyframe based on the number of structured lines in the current frame and the relative transformation with the previous vanishing point keyframe. We establish a set of vanishing point keyframes $\mathbb{F} = \{\mathbf{F}_i\}_{i=1}^n$ with successful vanishing point tracking. Using the incremental update principle, we synthesize multiple observed vanishing point keyframes to update global vanishing points V^{new}. We make the current global 3-D vanishing point left multiply a rotation matrix R_v and optimize this rotation matrix to minimize the Euclidean distance between the transformed 3-D vanishing point and the 3-D vanishing points V^i of all historical keyframes

$$\min_{R_v} \sum_{i=1}^{n} \left\|V^i - R_v V^{\text{new}}\right\|_2^2. \tag{17}$$

We use the Levenberg–Marquardt algorithm to iteratively solve this least squares problem to obtain the best update results for vanishing point.

B. Vanishing Point Measurement Residual

After vanishing point update, we translate the vanishing point v^g in global frame to v^c in current camera frame according to the property that the vanishing point transformation is only related to the rotation matrix in the same Manhattan world. The difference between the projection of global vanishing point and the observed vanishing point in the camera frame is the vanishing point observation error r_V, as shown in Fig. 5. Then, we construct (18) to indicate vanishing point observation error

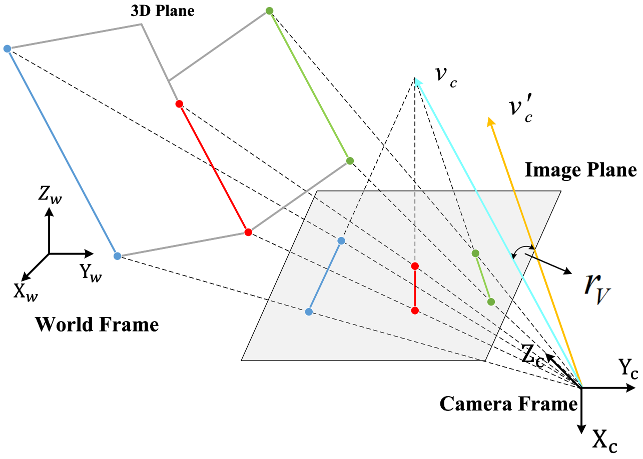


Fig. 5. Illustration of vanishing point measurement error.

$$r_V = v^c - (v^c)' = v^c - R_b^c R_g^b v^g. \tag{18}$$

We hope to minimize the vanishing point observation error by adjusting rotation matrix. In (18), the residual between observed vanishing point and the estimated vanishing point is the difference between the two normalized vectors. In order to solve the scale problem of the vanishing point measurement residual, observation error for vanishing point is further transformed into the following form:

$$r_V = a\cos\left(v^c \cdot \left(R_b^c R_g^b v^g\right)\right) \tag{19}$$

where the Jacobian matrix derived from r_V to the rotation matrix is

$$\frac{\partial r_V}{\partial \zeta} = \frac{\partial \arccos(\delta)}{\partial \zeta} = -\frac{1}{\sqrt{1-\delta^2}} v^c \left(R_b^c R_g^b v^g\right)_\times \tag{20}$$

where $(\cdot)_\times$ represents a 3 × 3 skew symmetric matrix

$$\delta = v^c \left(R_b^c R_g^b v^g\right). \tag{21}$$

In this algorithm, if detected vanishing points in the vanishing point keyframes, the residual error can be constructed.

C. Spatial Consistency of Vertical Lines

Through the vanishing point detection, we divide the image lines into three categories on the basis of Manhattan world hypothesis. Then, we use the gravity vector measured by IMU to separate the set of vertical lines. We select the long-tracked line segments from the group of vertical lines to construct the residual error by colinear constraints with gravity vectors. First, two lines in the set of vertical lines are represented by the plücker coordinate method as

$$\begin{cases} L_i^g = \left[n_i^g, v_i^g\right] \\ L_j^g = \left[n_j^g, v_j^g\right]. \end{cases} \tag{22}$$

We can translate these lines from global center to camera center

$$\begin{cases} L_i^c = \bar{T}_g^c L_i^g = \left[n_i^c, v_i^c\right] \\ L_j^c = \bar{T}_g^c L_j^g = \left[n_j^c, v_j^c\right]. \end{cases} \tag{23}$$

Then, we use the IMU measurement model for position and velocity to obtain the gravity vector at the zero frame of the camera. The IMU measurement model can be indicated as

$$\begin{cases} \alpha_{b_{k+1}}^{b_k} = R_{c_0}^{b_k}\left(sp_{b_{k+1}}^{c_0} - sp_{b_k}^{c_0} + \frac{1}{2}g^{c_0}\Delta t_k^2 - R_{b_k}^{c_0} v_{b_k}^{b_k}\Delta t_k\right) \\ \beta_{b_{k+1}}^{b_k} = R_{c_0}^{b_k}\left(R_{b_{k+1}}^{c_0} v_{b_{k+1}}^{b_{k+1}} + g^{c_0}\Delta t_k - R_{b_k}^{c_0} v_{b_k}^{b_k}\right) \end{cases} \tag{24}$$

where $\alpha_{b_{k+1}}^{b_k}$ and $\beta_{b_{k+1}}^{b_k}$ are position and velocity transformed, respectively, b_k and b_{k+1} are two consecutive frames in sliding window. $R_{c_0}^{b_k}$ is the rotation matrix between the initial camera coordinate and the body coordinate at frame k. $p_{b_{k+1}}^{c_0}$ and $p_{b_k}^{c_0}$ are translation vector between camera coordinate and the kth frame body coordinate. The variables to be optimized include g^{c_0}, s, and $v_{b_k}^{b_k}$. g^{c_0} is the gravity vector at initial camera coordinate and s aligns the visual structure with absolute scale. $v_{b_k}^{b_k}$ is the velocity in body frame at time b_k and Δt_k is the time interval between b_k and b_{k+1}.

Substitute the conversion relationship between the camera frame and the IMU frame into (24). Rewrite (24) using matrix form

$$\begin{aligned} &\begin{bmatrix} \alpha_{b_{k+1}}^{b_k} + R_{c_0}^{b_k} R_{b_{k+1}}^{c_0} p_c^b - p_c^b \\ \beta_{b_{k+1}}^{b_k} \end{bmatrix} \\ &\quad = H_{b_{k+1}}^{b_k} X_I + n_{b_{k+1}}^{b_k} \\ &\quad \cong R_{c_0}^{b_k} \begin{bmatrix} -{R_{c_0}^{b_k}}^{-1}\Delta t_k & 0 & \frac{1}{2}\Delta t_k^2 & p_{c_{k+1}}^{c_0} - p_{c_k}^{c_0} \\ -{R_{c_0}^{b_k}}^{-1} & R_{b_{k+1}}^{c_0} & \Delta t_k & 0 \end{bmatrix} \begin{bmatrix} v_{b_k}^{b_k} \\ v_{b_{k+1}}^{b_{k+1}} \\ g^{c_0} \\ s \end{bmatrix} \end{aligned} \tag{25}$$

where rough gravity vector g^{c_0} can be solved by matrix decomposition.

The gravity vector obtained from (25) can be refined through the known magnitude of the gravity vector. The gravity is reparametrized in two variables on the tangent space by denoting it as

$$G = \left\|g^{c_0}\right\| \hat{\bar{g}}^{c_0} + b_{3\times2} w_{2\times1} \tag{26}$$

where $\|g^{c_0}\|$ is the magnitude of gravity and G lies on a sphere with the radius $\|g^{c_0}\|$. $\hat{\bar{g}}^{c_0}$ is the current estimated direction vector and b is the orthogonal basis spanning the tangent plane. w is the corresponding displacements to b.

From the vanishing point initialization, it is known that the vertical set of parallel lines corresponding to the vanishing point is colinear with the gravity vector G

$$\begin{cases} v_i^c \otimes G = 0_{3\times1} \\ v_i^c \otimes v_j^c = 0_{3\times1} \\ v_i^T n_j^T + n_i^T v_j^T = 0_{3\times1}. \end{cases} \tag{27}$$

Because of the observation noise, these two parallel lines in experimental setting usually do not exactly satisfy the parallel regularity, so we provide a spatial consistency constraint on the parallel lines for pose estimation. The spatial consistency error of vertical parallel lines as follows:

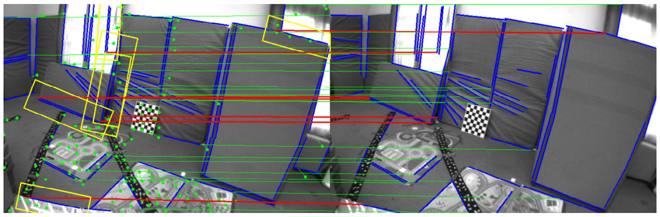


Fig. 6. Comparisons of our proposed algorithm and LBD line matching method based on ELSED line segment extraction algorithm. The blue lines indicate the line features extracted by ELSED. The green lines indicate the correspondence lines matched by the LBD algorithm. The red lines show the matched lines added after matching by the hybrid tracking algorithm. The yellow rectangular box is the search box for line segments, which is used to search for nearest neighbor points. The green dots are the feature points successfully matched by the optical flow method, and the motion direction of these points is shown.

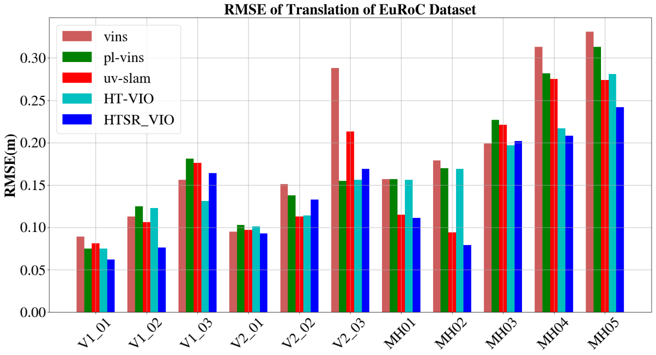


Fig. 7. RMSEs of translation on EuRoC dataset are shown.

$$r_S = v_i^c \otimes G + v_i^c \otimes v_j^c + v_i^T n_j^T + n_i^T v_j^T. \tag{28}$$

Relying on the chain rule, the Jacobian matrix of r_S is

$$J_S = \begin{bmatrix} \dfrac{\partial r_S}{\partial L_i^c}\left[\dfrac{\partial L_i^c}{\partial L_i^g}\cdot\dfrac{\partial L_i^g}{\partial o_i}, \dfrac{\partial L_i^c}{\partial \delta x}\right] \\ \dfrac{\partial r_S}{\partial L_j^c}\left[\dfrac{\partial L_j^c}{\partial L_j^g}\cdot\dfrac{\partial L_j^g}{\partial o_j}, \dfrac{\partial L_j^c}{\partial \delta x}\right] \end{bmatrix}^T. \tag{29}$$

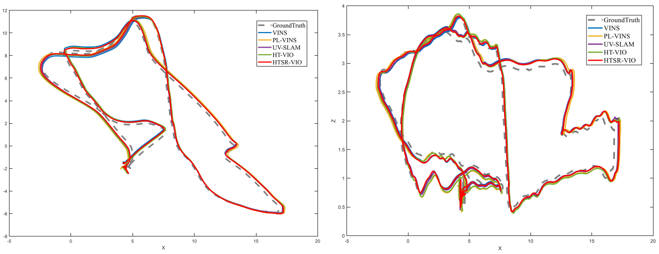
V. Experimental Results

The experiments of whole system were performed with an Intel Core i7-10700K CPU with 16 GB of memory. We evaluated our proposed method on UAVs public datasets of EuRoC, Kaist-VIO and NTU Viral with visual and inertial information, which were acquired using rotorcraft with camera and IMU, enabling synchronized visual and inertial measurements. These datasets include indoor scenes in machine halls, offices, laboratories, and outdoor scenes on campus with ground truth trajectories and accurate external parameters between the camera and IMU. In our work, we leveraged the provided parameters and data to perform comparison experiments.[1]

[1] The video of experiments is available at http://aius-lab.net/htsr-vio-real-time-line-based-visual-inertial-odometry-with-point-line-hybrid-tracking-and-structural-regularity/.

TABLE I
COMPARISON OF THE NUMBER OF LONG-TERM TRACKING LINE FEATURES BY LBD AND HYBRID TRACKING ALGORITHM ON EUROC DATASET

Sequence	ELSED+LBD	ELSED+Ours
V1_01_easy	28730	**30131**
V1_02_medium	14107	**17421**
V1_03_difficult	10032	**11545**
V2_01_easy	20822	**22413**
V2_02_medium	18040	**19846**
V2_03_difficult	10155	**11094**
MH_01_easy	23100	**25189**
MH_02_easy	15387	**17457**
MH_03_medium	20188	**22359**
MH_04_difficult	17332	**18574**
MH_05_difficult	17457	**19273**



(a) (b)

Fig. 8. Trajectory comparison of experiments on MH_05 sequence of EuRoC dataset. Where the dashed line represents the ground truth and several other colors represent the SOTA work and our proposed algorithm, respectively. (a) and (b) represent the top view of *xy*-axis and the side view of *xz*-axis, respectively.

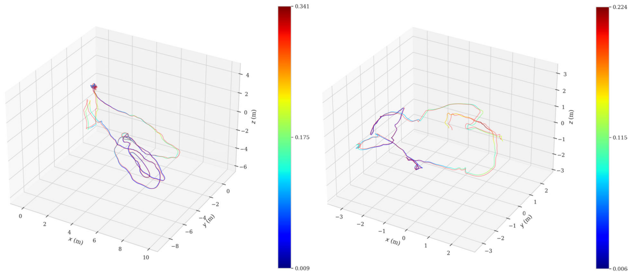


Fig. 9. Trajectory error between our proposed method with reference.

A. Evaluation of Hybrid Tracking Algorithm

To evaluate the proposed hybrid tracking algorithm, we used long-term observed lines for tracking to avoid outlier interference. Then, a mask similar to nonmaxima suppression was applied to make the feature point distribution both dense and uniform, which facilitated line feature prediction using feature points. In our experiment, the number of line segments that were observed multiple times was used as a metric to evaluate the performance of line tracking. The comparisons of our proposed algorithm and LBD line matching method based on ELSED line segment extraction algorithm are shown in Fig. 6. Clearly, our proposed method achieves better matching performances than LBD-only method. In Table I, the results of long-term tracked lines number are shown, and our proposed line segments tracking algorithm can track more reliable lines than classical method.

B. Localization Accuracy Evaluation

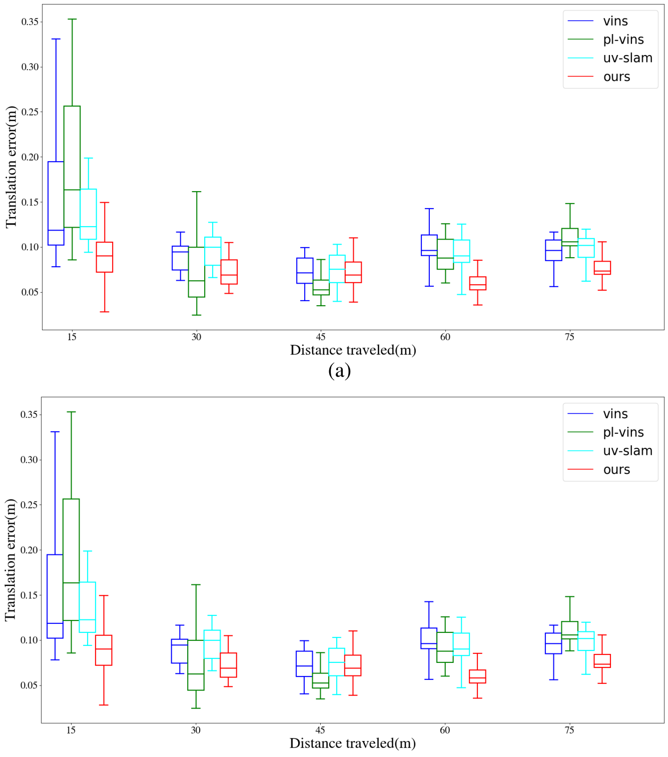
To evaluate the accuracy of the proposed system, we measured the root mean square error (RMSE) of translation on the EuRoC, Kaist-VIO, and NTU-Viral datasets for our proposed method and three other methods: VINS-mono, PL-VINS, and UV-SLAM. We fellow the default experimental parameters given by authors of three methods. The results indicate that the proposed method outperforms these state-of-the-art (SOTA) methods in Fig. 7.

In Table II, the values of the RMSE of the absolute position error for each method evaluated on the EuRoC, Kaist-VIO, and NTU-Viral dataset are shown in detail. Due to space limitation, we abbreviate the names of data series in the dataset, for example, Circle_fast in Kaist-VIO dataset is abbreviated as Cir_f, and eee_01 in NTU-Viral is abbreviated as E_1. In this table, HT-VIO refers to the visual-inertial odometry with hybrid tracking algorithm and LPB residual constraint only. Experimental results on the EuRoc dataset show HTSR-VIO is generally superior in machine hall and simple office environments such as the MH_01-MH_05 and the V1_01, V1_02, V2_01 data sequences. In the machine room environment where lines are neat and consistent with the Manhattan world assumption, the spatial consistency constraints on the vanishing point observation residuals and vertical lines can effectively improve the performance of our algorithm. On the other hand, in more complex indoor environments with cluttered line segments, such as the V1_03, V2_02, V2_03 data sequences, the correct detection of vanishing points becomes more challenging. In these cases, the measurement model of vanishing points may carry incorrect structural information, which can interfere with the system state estimation and result in accuracy degradation. The experimental results on the Kaist-VIO dataset show that HTSR-VIO performs better than HT-VIO without adding structural line constraints in the case of dramatic exercise, such as the Cir_f, Cir_h, Rot_n, and Rot_f data sequences. In the scenario of uniform motion (Cir_n, Squ_n), the performance of HTSR-VIO is not significantly different from UV-SLAM, PL-VINS, and VINS-mono. Overall, excluding the failure of each algorithm, HTSR-VIO indicates an accuracy improvement of 27.2% and 18.7% over VINS-mono, 20.5% and 39.2% over PLVINS, and 13.1% and 39.5% over UV-SLAM on the EuRoC and Kaist-VIO dataset, respectively.

The data sequence of NTU-Viral was acquired in a large outdoor scene, which is an Atlanta world consisting of multiple Manhattan worlds, and thus has multiple different sets of vanishing points. This is not consistent with the Manhattan world assumed in this article. To avoid this situation as much as possible, we select the E_1, E_2, and E_3 sequences of the NTU-Viral dataset that are more consistent with the Manhattan assumption for our experiments. During the experiments, the weights of the vanishing point observation residual were reduced, but the IMU-constrained spatially consistent residuals were still used normally. The experimental results show that HTSR-VIO performs better in the E_2 sequence only, while HT-VIO performs better in the other two sequences, which is in accordance with our expectation.

TABLE II
RMSE RESULTS ON EUROC, KAIST-VIO, AND NTU-VIRAL MAV DATASET. (UNIT: METER)

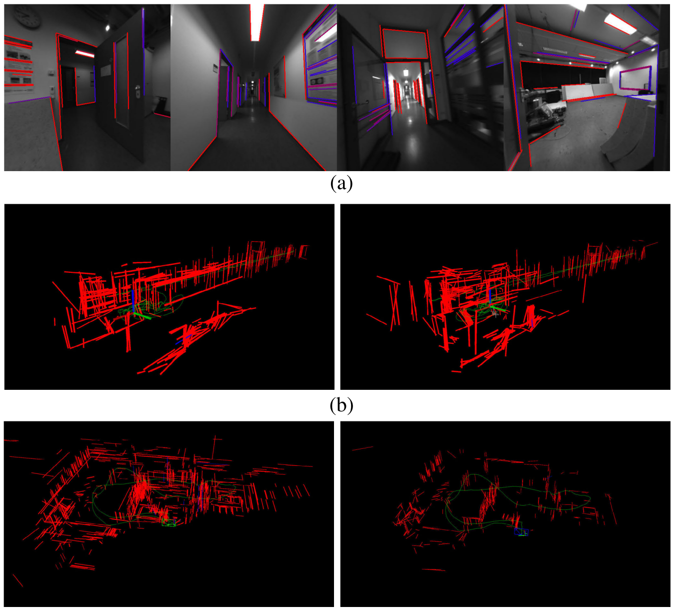
	EuRoc											Kaist-VIO											NTU-Viral			Best
	V1_1	V1_2	V1_3	V2_1	V2_2	V2_3	MH_1	MH_2	MH_3	MH_4	MH_5	Cir_n	Cir_f	Cir_h	Squ_n	Squ_f	Squ_h	Inf_n	Inf_f	Inf_n	Rot_n	Rot_f	E_1	E_2	E_3	Count
VINS-mono	0.089	0.113	0.156	0.095	0.151	0.288	0.157	0.179	0.199	0.313	0.331	**0.105**	0.132	0.361	0.066	0.070	fail	**0.050**	**0.065**	0.457	fail	0.375	2.22	0.565	0.624	3
PL-VINS	0.075	0.125	0.181	0.103	0.137	**0.155**	0.157	0.170	0.227	0.282	0.313	0.111	0.156	0.429	0.071	0.096	0.294	0.091	0.107	0.596	0.790	fail	2.37	0.674	0.692	1
UV-SLAM	0.081	0.106	0.176	0.097	**0.113**	0.213	0.115	0.094	0.221	0.275	0.274	0.109	0.148	0.645	**0.061**	0.079	fail	**0.050**	0.127	0.582	fail	fail	fail	fail	fail	3
HT-VIO (ours)	0.075	0.123	**0.131**	0.101	0.114	0.156	0.156	0.169	**0.197**	0.217	0.281	0.110	0.148	0.482	0.065	**0.069**	**0.267**	0.064	0.087	0.579	0.617	fail	**2.13**	0.643	**0.622**	6
HTSR-VIO (ours)	**0.062**	**0.076**	0.164	**0.093**	0.133	0.169	**0.111**	**0.079**	0.202	**0.208**	**0.242**	0.110	**0.117**	**0.277**	0.067	0.075	0.283	0.087	0.066	**0.289**	**0.294**	**0.278**	2.31	**0.483**	0.634	13



(a)

(b)

Fig. 10. Translation error and rotation error on EuRoC dataset. (a) Translation error. (b) Rotation error.



(a)

(b)

(c)

Fig. 11. Results of mapping and trajectory on corridor4 sequence of TUM-6 dataset and MH_04 sequence of EuRoC dataset. (a) Blue lines indicate detected lines for the first time and the red lines indicate long-term tracking lines. Line features with trace counts between 0 and 5 are represented by colors between blue and red. The left image in (b) and (c) show the line maps reconstructed by HTSR-VIO in TUM-6 dataset and EuRoC dataset. The right images are line maps of PL-VINS.

Then, our proposed algorithm is compared with other methods for the trajectory term as shown in Fig. 8. The results show that the trajectory of HTSR-VIO is closer to the ground truth, i.e., the accuracy of our proposed algorithm is higher than that of other algorithms. In order to show the trajectories error of our proposed method with reference, the flight paths of the UAVs in the MH_01 and V2_01 sequences were visualized as shown in Fig. 9. Among them, the trajectories on the XZ plane are shown to differentiate. To test the proposed system in more depth, we employed relative translation error and rotation error to check the local accuracy of the trajectories. As illustrated in Fig. 10, the translation errors produced by our method are smaller and show smaller fluctuations compared to several other methods. In terms of the evaluation of rotation error, our rotation error is large at the initial stage, which may be due to the delayed initialization of the vanishing point. However, as seen in several subsequent data series, our rotation error gradually decreases, demonstrating the superiority of the vanishing point in reducing accumulated rotation error.

Then, the reconstruction line maps in corridor4 sequence of TUM-6 dataset [37] and MH_04_difficulty sequence of EuRoC dataset are depicted in Fig. 11, where the red lines in line maps represent line features. The line maps demonstrate that, in machine hall and corridor scenes with abundant line features, an excellent structured line map can be established using line segments. Due to incomplete ground truth of TUM-VI dataset, we do not leverage this dataset for experiments on localization accuracy.

C. Evaluation of System Runtime

The addition of line detector and vanishing point tracking module in the front end of VIO increases both the time consumption and the computational complexity. The average tracking time of the line segment tracking thread is 20.66 ms, which is less time-consuming than UV-SLAM. In Fig. 12,

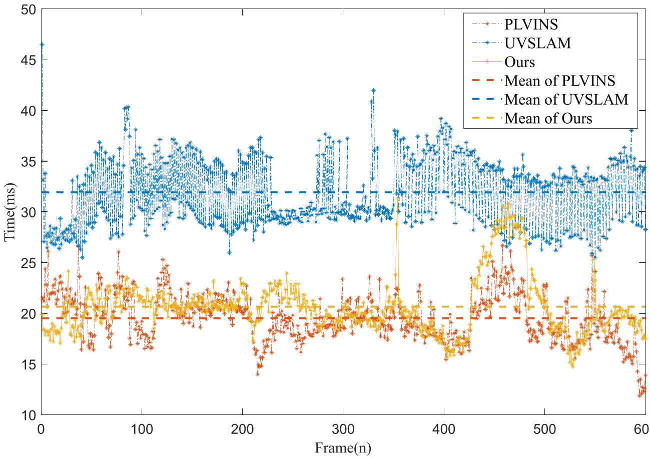


Fig. 12. Time cost per frame for feature tracking of PLVINS, UV-SLAM, and our proposed method.

TABLE III
AVERAGE RUNTIME COMPARISON. (UNIT: MILLISECOND)

Module	Times			
	VINS-mono	PL-VINS	UV-SLAM	ours
Point Tracking	16.35	16.51	16.42	17.03
Line Extraction	-	11.21	7.23	5.58
Line Tracking	-	19.51	31.93	20.66
VP Detection	-	-	9.31	1.27
Back End	42.35	45.85	102.56	62.27

the proposed vanishing point update mechanism significantly outperforms UV-SLAM in efficiency, and the time consumption is almost the same as PL-VINS($\sim$1 ms) as the time consumption of the PL-VINS algorithm. Moreover, from the time consumption shown in Table III, we observe that the time cost of backend is increased by only a dozen milliseconds. Experimental results show that our system can run in real time.

VI. CONCLUSION

In this article, a novel visual inertial estimator with point-line hybrid tracking algorithm and structural regularity was proposed to obtain accuracy motion poses. On one hand, we proposed a hybrid tracking algorithm which utilized short-term line prediction based on the matched points in the target region and their pixel velocity, and neighbor line matching based on reverse verification and gray check. Then, LPB residual error model was constructed based on the distance between the tracked line feature and its neighbor points to refine the estimation of landmarks and poses. On the other hand, we designed an efficient vanishing point update mechanism to eliminate the incorrect vanishing point caused by noise and interfere, and established vanishing point keyframe strategy to reduce the update frequency. To cope with the rotation drift and position offset, we constructed vanishing points global observation error model and spatial geometric consistency between vertical lines and gravity vector. Experimental results show that the proposed hybrid tracking algorithm effectively improved line segment tracking results, and our proposed PLB residual error and spatial geometric consistency constraint refined the positional estimation and improved the localization accuracy. Furthermore, we tested the proposed algorithm on the EuRoC dataset and compared with the SOTA algorithm. The results show that the proposed system can operate in real time and had better accuracy than above SOTA methods. In future work, we will explore geometric information detected by multisensors to accurately construct accurate geometric maps, and combine semantic information to build interactive maps.

REFERENCES

[1] A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse, "MonoSLAM: Real-time single camera SLAM," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 6, pp. 1052–1067, Jun. 2007.

[2] K. Jung, Y. Kim, H. Lim, and H. Myung, "ALVIO: Adaptive line and point feature-based visual inertial odometry for robust localization in indoor environments," in *Proc. Int. Conf. Robot. Intell. Technol.*, 2020, pp. 171–184.

[3] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardós, "ORB-SLAM: A versatile and accurate monocular SLAM system," *IEEE Trans. Robot.*, vol. 31, no. 5, pp. 1147–1163, Oct. 2015.

[4] J. Engel, T. Schops, and D. Cremers, "LSD-SLAM: Large-scale direct monocular SLAM," in *Proc. 13th Eur. Conf. Comput. Vis. (ECCV)*, Zurich, Switzerland. Cham, Switzerland: Springer, 2014, pp. 834–849.

[5] G. Klein, D. Murray, and J. D. Tards, "Parallel tracking and mapping for small AR workspaces," in *Proc. IEEE ACM Int. Symp. Mixed Augmented Reality (ISMAR)*, Nara, Japan, Nov. 2007, pp. 225–234.

[6] H. Yin, S. Li, Y. Tao, J. Guo, and B. Huang, "Dynam-SLAM: An accurate, robust stereo visual-inertial SLAM method in dynamic environments," *IEEE Trans. Robot.*, vol. 39, no. 1, pp. 289–308, Feb. 2023.

[7] P. Chen, W. Guan, and P. Lu, "ESVIO: Event-based stereo visual inertial odometry," *IEEE Robot. Autom. Lett.*, vol. 8, no. 6, pp. 3661–3668, Jun. 2023.

[8] R. Mur-Artal and J. D. Tardós, "ORB-SLAM2: An open-source SLAM system for monocular, stereo, and RGB-D cameras," *IEEE Trans. Robot.*, vol. 33, no. 5, pp. 1255–1262, Oct. 2017.

[9] J. Engel, V. Koltun, and D. Cremers, "Direct sparse odometry," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 3, pp. 611–625, Mar. 2018.

[10] 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.

[11] X. Liu et al., "Fast eye-in-hand 3D scanner-robot calibration for low stitching errors," *IEEE Trans. Ind. Electron.*, vol. 68, no. 9, pp. 8422–8432, Sep. 2021.

[12] J. Fu, G. Sun, W. Yao, and L. Wu, "On trajectory homotopy to explore and penetrate dynamically of multi-UAV," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 12, pp. 24008–24019, Dec. 2022.

[13] R. Gomez-Ojeda, F.-A. Moreno, D. Zuniga-Noel, 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.

[14] 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, pp. 1–25, 2018.

[15] Q. Fu et al., "PL-VINS: Real-time monocular visual-inertial SLAM with point and line features," 2020, *arXiv:2009.07462*.

[16] 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.

[17] J. Lee and S.-Y. Park, "PLF-VINS: Real-time monocular visual-inertial SLAM with point-line fusion and parallel-line fusion," *IEEE Robot. Autom. Lett.*, vol. 6, no. 4, pp. 7033–7040, Oct. 2021.

[18] F. Shu, J. Wang, A. Pagani, and D. Stricker, "Structure PLP-SLAM: Efficient sparse mapping and localization using point, line and plane for monocular, RGB-D and stereo cameras," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2023, pp. 2105–2112.

[19] L. Zhou, G. Huang, Y. Mao, S. Wang, and M. Kaess, "EDPLVO: Efficient direct point-line visual odometry," in *Proc. Int. Conf. Robot. Autom. (ICRA)*, May 2022, pp. 7559–7565.

[20] Y. Zhao and P. A. Vela, "Good line cutting: Towards accurate pose tracking of line-assisted VO/VSLAM," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 516–531.

[21] Q. Wang, Z. Yan, J. Wang, F. Xue, W. Ma, and H. Zha, “Line flow based simultaneous localization and mapping,” *IEEE Trans. Robot.*, vol. 37, no. 5, pp. 1416–1432, Oct. 2021.

[22] H. Wei, F. Tang, C. Zhang, and Y. Wu, “Highly efficient line segment tracking with an IMU-KLT prediction and a convex geometric distance minimization,” in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2021, pp. 3999–4005.

[23] L. Xu, H. Yin, T. Shi, D. Jiang, and B. Huang, “EPLF-VINS: Real-time monocular visual-inertial SLAM with efficient point-line flow features,” *IEEE Robot. Autom. Lett.*, vol. 8, no. 2, pp. 752–759, Feb. 2023.

[24] J.-C. Bazin and M. Pollefeys, “3-line RANSAC for orthogonal vanishing point detection,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2012, pp. 4282–4287.

[25] J.-C. Bazin et al., “Globally optimal line clustering and vanishing point estimation in Manhattan world,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2012, pp. 638–645.

[26] Y. H. Lee, C. Nam, K. Y. Lee, Y. S. Li, S. Y. Yeon, and N. L. Doh, “VPass: Algorithmic compass using vanishing points in indoor environments,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Oct. 2009, pp. 936–941.

[27] D. Zou, Y. Wu, L. Pei, H. Ling, and W. Yu, “StructVIO: Visual-inertial odometry with structural regularity of man-made environments,” *IEEE Trans. Robot.*, vol. 35, no. 4, pp. 999–1013, Aug. 2019.

[28] B. Xu, P. Wang, Y. He, Y. Chen, Y. Chen, and M. Zhou, “Leveraging structural information to improve point line visual-inertial odometry,” *IEEE Robot. Autom. Lett.*, vol. 7, no. 2, pp. 3483–3490, Apr. 2022.

[29] H. Lim, J. Jeon, and H. Myung, “UV-SLAM: Unconstrained line-based SLAM using vanishing points for structural mapping,” *IEEE Robot. Autom. Lett.*, vol. 7, no. 2, pp. 1518–1525, Apr. 2022.

[30] M. Burri et al., “The EuRoC micro aerial vehicle datasets,” *Int. J. Robot. Res.*, vol. 35, no. 10, pp. 1157–1163, Sep. 2016.

[31] J. Jeon, S. Jung, E. Lee, D. Choi, and H. Myung, “Run your visual-inertial odometry on NVIDIA Jetson: Benchmark tests on a micro aerial vehicle,” *IEEE Robot. Autom. Lett.*, vol. 6, no. 3, pp. 5332–5339, Jul. 2021.

[32] T.-M. Nguyen, S. Yuan, M. Cao, Y. Lyu, T. H. Nguyen, and L. Xie, “NTU VIRAL: A visual-inertial-ranging-LiDAR dataset, from an aerial vehicle viewpoint,” *Int. J. Robot. Res.*, vol. 41, no. 3, pp. 270–280, Mar. 2022.

[33] J. Shi and C. Tomasi, “Good features to track,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 1994, pp. 593–600.

[34] I. Suárez, J. M. Buenaposada, and L. Baumela, “ELSED: Enhanced line segment drawing,” *Pattern Recognit.*, vol. 127, Jul. 2022, Art. no. 108619.

[35] 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, Oct. 2013.

[36] H. Li, J. Zhao, J.-C. Bazin, and Y.-H. Liu, “Quasi-globally optimal and near/true real-time vanishing point estimation in Manhattan world,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 3, pp. 1503–1518, Mar. 2022.

[37] 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.

Bing Zhang (Graduate Student Member, IEEE) received the B.S. degree from North Central University, Taiyuan, China, in 2017, and the M.S. degree from Harbin Institute of Technology, Harbin, China, in 2020, where he is currently pursuing the Ph.D. degree with the Department of Control Science and Engineering.

His research interests include multimodal feature fusion in perception and mapping, such as multisource localization systems for mobile robots, 3-D reconstruction, and visual simultaneous localization and mapping (VSLAM).

Yankun Wang (Graduate Student Member, IEEE) received the B.S. degree from Northeastern University, Shenyang, China, in 2016. He is currently pursuing the Ph.D. degree with the Department of Control Science and Engineering, Harbin Institute of Technology University, Harbin, China.

His research interests include multisensor fusion in terms of navigation and mapping technologies such as inertial navigation, simultaneous localization and mapping (SLAM), and mobile multisensor mapping systems.

Weiran Yao (Member, IEEE) received the bachelor’s (Hons.), master’s, and Ph.D. degrees in aeronautical and astronautical science and technology from the School of Astronautics, Harbin Institute of Technology (HIT), Harbin, China, in 2013, 2015, and 2020, respectively.

From 2017 to 2018, he was a Visiting Ph.D. Student with the Department of Mechanical and Industrial Engineering, University of Toronto (UofT), Toronto, ON, Canada. He is currently an Associate Professor with the School of Astronautics, HIT. His research interests include multirobot systems, and perception and control.

Guanghui Sun (Senior Member, IEEE) received the B.S. degree in automation and the M.S. and Ph.D. degrees in control science and engineering from Harbin Institute of Technology, Harbin, China, in 2005, 2007, and 2010, respectively.

He is currently a Professor with the Department of Control Science and Engineering, Harbin Institute of Technology. His research interests include autonomous intelligent unmanned systems, fractional order systems, nonlinear control systems, and sliding mode control.