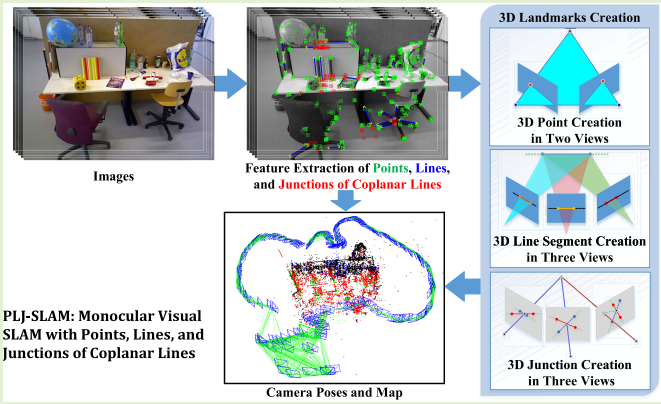
PLJ-SLAM: Monocular Visual SLAM With Points, Lines, and Junctions of Coplanar Lines

Guangli Ren, Zhiqiang Cao, *Senior Member, IEEE*, Xilong Liu, Min Tan, and Junzhi Yu, *Fellow, IEEE*

***Abstract*—Existing monocular visual simultaneous localization and mapping (SLAM) mainly focuses on point and line features, and the constraints among line features are not fully explored. In this paper, a multi-feature monocular SLAM with ORB points, lines, and junctions of coplanar lines is proposed for indoor environments. To create 3D junctions of coplanar lines, an adaptive coordinate confidence is designed to describe the coordinate stability of 2D junctions on the image plane. Based on the matched 2D junctions between the current keyframe and other two associated keyframes, a multi-view coplanarity verification is performed for 3D junction creation. Moreover, reprojection verification is conducted to update the observation of 3D junction, which is used to dynamically update the confidence of coplanarity for confidence-based local bundle adjustment. The camera pose is optimized through multiple constraints of ORB points, lines, and junctions of the coplanar lines. As a result, the performance of monocular SLAM is improved. The experiment verification shows the effectiveness of the proposed method.**



***Index Terms*—Junction of coplanar lines, confidence of coplanarity, multi-view, multi-feature monocular SLAM.**

I. Introduction

IN RECENT years, simultaneous localization and mapping (SLAM) has received much attention in the fields of robot localization and autonomous driving. LiDAR SLAM and visual SLAM are two mainstream branches. The former calculates the relative motion and pose changes of the LiDAR sensor by point cloud matching [1], [2], and precise depth measurements provide the potential for accurate pose estimation. However, this type of sensor is relatively expensive. Also, the environment representation through point cloud is relatively limited. Compared with LiDAR, the visual sensors enjoy the advantages in size, weight, and price with abundant information, which makes visual SLAM more popular [3]. For the visual SLAM, the pose of the visual sensor can be estimated by image feature extraction and matching.

In the early stage of visual SLAM, limited by computing resources, filter-based methods were mainly used [4]. However, complex and large-scale scenarios inevitably increase the feature amount that needs to be processed. In this case, the efficiency of the filter-based SLAM is affected. To solve this problem, researchers began to learn from graph optimization theory in structure from motion (SFM) [5], especially bundle adjustment (BA). The computational cost of the initial BA is high, which affects the real-time performance of visual SLAM with a large number of features and camera poses to be optimized. The concept of keyframe as well as graph optimization acceleration schemes was then proposed [5]–[8]. These methods can be further divided into two categories: direct and feature-based methods. The former [9]–[11] uses all pixels to estimate camera pose with the assumption of gray scale invariant, which is robust to motion blur, however, illumination changes can affect its performance. The latter achieves pose estimation with sparse features by optimizing the reprojection error between frames. This solution has better robustness, and typical methods include ORB-SLAM [6] and PL-SLAM [8]. Point feature whose local texture changes significantly is commonly used. As a kind of continuous geometric element, the line feature is also adopted with its rich structural description. It should be

Manuscript received 4 June 2022; accepted 12 June 2022. Date of publication 27 June 2022; date of current version 1 August 2022. This work was supported in part by the National Natural Science Foundation of China under Grant 62073322, Grant 61633020, and Grant 61836015. The associate editor coordinating the review of this article and approving it for publication was Dr. Ioannis Raptis. *(Corresponding author: Xilong Liu.)*

Guangli Ren, Zhiqiang Cao, Xilong Liu, and Min Tan are with the State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China, and also with the School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing 100049, China (e-mail: renguangli2018@ia.ac.cn; zhiqiang.cao@ia.ac.cn; xilong.liu@ia.ac.cn; min.tan@ia.ac.cn).

Junzhi Yu is with the State Key Laboratory for Turbulence and Complex System, Department of Mechanics and Engineering Science, BIC-ESAT, College of Engineering, Peking University, Beijing 100871, China (e-mail: junzhi.yu@ia.ac.cn).

Digital Object Identifier 10.1109/JSEN.2022.3185122

pointed out that these two features are not opposites. Many researches combine point and line features to improve performance of the system [8], [12]–[17], and they mainly add constraints based on line endpoints. Qian *et al.* proposed a visual SLAM using egocentric stereo sensor to ensure accurate and robust dynamic egolocalization performance, where bags of point and line word pairs are designed in loop closure detection [12]. This method has been successfully applied for wearable-assisted substation inspection with good practicability. Upon RGB-D sensor, Fu *et al.* proposed a complete high-accuracy SLAM based on a combination of points and lines, which achieves better adaptability to low texture indoor environments [13]. Actually, there also exist geometric constraints for line pairs. If these geometric constraints are further explored, the SLAM system with better performance is expected.

In the field of projective geometry, the intersection of lines is a common phenomenon, and the junction of two lines often implies geometric constraints. As Kim and Lee pointed out in [18], the junctions of coplanar lines have been proven to have favorable localization property. By introducing the junctions of coplanar lines as new features into the visual SLAM, the features can be further enriched, which is equivalent to that the coplanar constraints of lines are implicitly added. And it is beneficial to improve the accuracy of pose estimation. The extraction of the coplanar lines' junctions is still challenging, especially for monocular SLAM. This involves not only the data association of the same line feature among multiple frames but also the coplanar relationship between different lines.

In this paper, we exploit the application of junctions of coplanar lines with the combination of points and lines for a monocular SLAM system in indoor environments. The contributions are threefold:

- A novel monocular SLAM using points, lines, and junctions of coplanar lines is proposed. This solution aims to improve performance by combining multiple features.
- 3D junctions of coplanar lines are created in the local mapping based on matching pairs of 2D junctions with coplanarity verification in multiple views.
- A unified optimization model is constructed to concurrently minimize reprojection errors of point, line, and junction of coplanar lines in the bundle adjustment, where the confidence of coplanarity is employed to calculate covariance matrices of 3D junctions.

The remainder of the paper is organized as follows. Section II details related work. The system overview is presented in Section III. Section IV describes the proposed method. The experiments are provided in Section V and Section VI concludes the paper.

II. Related Work

Visual SLAM uses the image sequence collected by visual sensor to estimate the pose of the camera while establishing an environmental map. In the following, monocular visual SLAM and the application of junctions of coplanar lines are reviewed.

A. Monocular Visual SLAM

Davison *et al.* proposed a real-time monocular SLAM (MonoSLAM) [4], which performs probabilistic modeling of the camera pose and 3D point position based on the extended Kalman filter. Due to the fact that it needs to estimate and update the states of the camera and all 3D points in the map, MonoSLAM can mainly be applied to a small scene. Unlike MonoSLAM that uses a single thread to update the camera pose and map framewisely, Klein and Murray simplified frame matching by designing the keyframe mechanism to reduce computing cost. And then parallel tracking and mapping (PTAM) [5] is achieved. Also, the use of BA optimization makes it achieve higher accuracy.

Since the idea of keyframe was proposed in [5], fruitful outcomes have ensued. Engel *et al.* proposed a large-scale direct monocular SLAM (LSD-SLAM) [10] that directly processes pixel features with a large intensity gradient. With the gradient-based pixel filtering, it constructs a large-scale consistent semi-dense map under the brightness constancy assumption. This assumption is overcome in direct sparse odometry (DSO) [11] through a full photometric calibration considering exposure time, lens vignetting, and non-linear response functions. Besides, point-based SLAMs have been proposed [6], [19], [20]. With features from accelerated segment test (FAST) [21], Herrera *et al.* proposed deferred triangulation visual SLAM (DT-SLAM) [19], which incrementally tracks individual 2D features until sufficient baseline for triangulation. This mechanism improves the stability of tracking even with pure rotation. Different from DT-SLAM that uses a local patch of pixels to describe a feature, ORB-SLAM adopts the oriented FAST and rotated BRIEF (ORB) [22] to describe point features [6]. It integrates automatic initialization and loop closure detection with the PTAM framework, which performs well in speed and map accuracy. ORB-SLAM2 [7] is an extended version that supports monocular, stereo, and RGB-D cameras. In some low texture environments with few point features, the pose estimation is sometimes unstable due to insufficient feature matching. To handle this problem, an edge point-based SLAM (EDGE-SLAM) [20] was proposed by applying edge point detection and matching refinement based on three views instead of traditional two-view feature extraction. Recently, Li *et al.* proposed an attention-based visual SLAM that simulates human navigation by combining a visual saliency model with traditional monocular visual SLAM [23]. Moreover, a new optimization method termed as weighted bundle adjustment is presented, which efficiently reduces the uncertainty of pose estimation.

In practice, the point map constructed by point-based SLAM does not sufficiently express the structural properties of the environment. Some researchers turn to visual SLAM using the line features instead of point features. Smith *et al.* conducted a line-based monocular extended Kalman filter SLAM [24]. A fast line detector is provided, where the candidate line segments generated by connecting FAST corners are hypothesized and tested. In some line-based SLAMs, the vanishing point is often employed to refine line features [25], [26] by constraining the direction of the lines. Also, loop detection [27], place recognition [28], and the combination of corner features

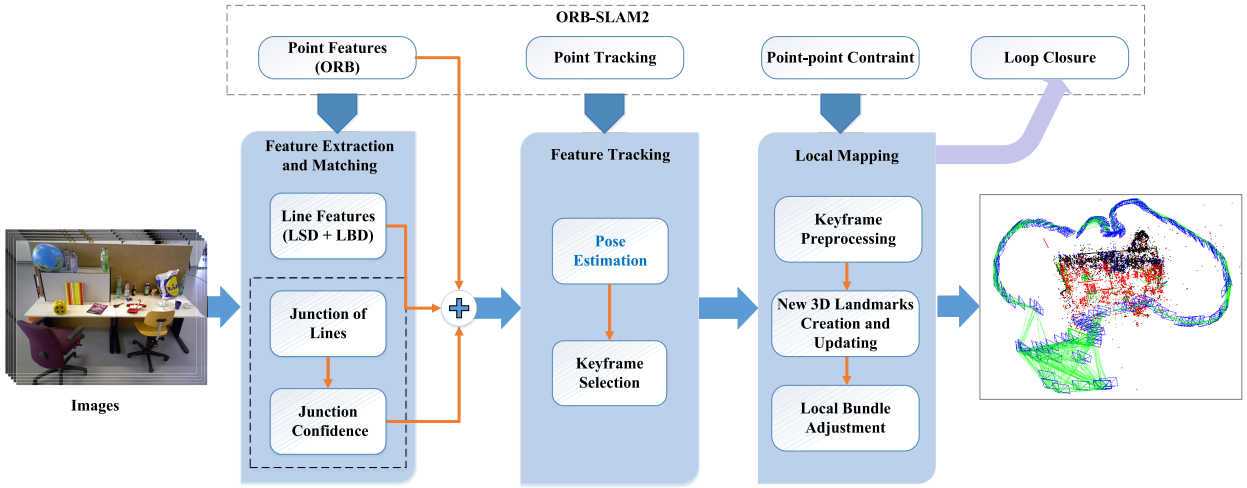


Fig. 1. Overall framework of the monocular visual PLJ-SLAM approach.

[29] are realized by using the line feature with its structural properties. In addition, a line-based SLAM with optimization is presented [30], where the plücker representation and cayley representation are used for line projection and optimization, respectively.

Overall, line-based SLAMs usually require structured scenes, which possibly affect their applications. The combination of point and line features becomes a natural choice. In [17], a geometric interpolation relying on epipolar geometry is designed to achieve point tracking in all frames for matrix factorization-based initialization, and localization is conducted with 3D lines by plücker and orthonormal representations. This solution is robust in some challenging environments. Another point-line-based work focuses on the initialization of visual odometry with small motion [31], where the point and plücker line reprojection constraints are adopted to estimate and optimize rotation with line segment correspondences, which provides good initial guesses for rotation and line directions. Pumarola *et al.* proposed a monocular visual SLAM based on point and line features (PL-SLAM) [8]. Defining the line segment by its two endpoints, PL-SLAM introduces the line segments satisfying three-view constraints into the optimization with the point feature to improve the accuracy of camera pose estimation as well as the adaptability to the low-texture environment.

B. Application of Junctions of Coplanar Lines

Most of the monocular SLAM methods based on point and line features use the endpoints of the line to establish matching to improve the accuracy of pose estimation. It is noted that some important characteristics of lines such as coplanar are also valuable, which deserves to be further researched. In the field of computer vision including 3D scene modeling [32], [33], feature correspondence [34], [35], and stereo visual SLAM [36], the junction of coplanar lines is used. In [32], adjacent lines' angles and their junction coordinate are integrated to form a line intersection context feature (LICF). And then simultaneous camera geometry estimation and line matching are conducted, where the LICF is matched by normalized cross-correlation (NCC) in two-view. Li *et al.* achieved line matching by designing a descriptor that represents the local region covered by two concentric circles centered at the junction [33]. With known epipolar geometry, Vincent and Laganiere solved point correspondences between widely separated views by estimating the local perspective distortion between the neighborhoods of junctions [34]. As for line-based stereo visual SLAM [36], junctions are described by rotated BRIEF for line matching, and an environment map with only line landmarks is constructed. An important usage of the junction feature is that it acts as an auxiliary mean of line matching [32], [33], [36]. Also, it is usually extracted in two-view and seldomly regarded as a 3D landmark in the environment map. In this paper, the extracted candidate junctions of coplanar lines in two-view are further verified incrementally in three views, and line matching serves for junction matching. On this basis, junctions are triangulated as landmarks for environment mapping.

III. System Overview

Existing point-line-based visual SLAMs mainly use the endpoints and direction of line segments to fuse with points, and the coplanarity of line segment pairs is not considered. In this paper, the junction of coplanar lines is introduced through the distance constraint of the spatial lines in multiple views. Thus a multi-feature monocular visual SLAM with point, line, and junction of coplanar lines is proposed for indoor environments, which is termed as PLJ-SLAM.

The pipeline of PLJ-SLAM is shown in Fig. 1. On the basis of ORB-SLAM2 method [7], points, lines, and junctions of coplanar lines are integrated to form a multi-feature framework. Given a stream of images, ORB points and line segments are extracted firstly, where the latter is obtained using the line segment detector (LSD) [37] and described by line binary descriptor (LBD) [38]. In addition, junctions on the image plane are formed by expanding the line segment along its direction. Considering that the accuracy of junction will decrease with the increasing of the distance between the junction and the line segment, coordinate confidence for each

junction is designed according to this distance. Then, not only the ORB point but also the line feature and junction matched with the local map are all used to estimate camera pose in the tracking stage. Also, the variation of the number of line segments and junctions is involved in keyframe selection for better data association. The generated keyframe will be sent to the local mapping, which further refines the current camera pose and updates the map. Firstly, the 3D landmarks of the current keyframe are added to a queue for stability verification. And then the verified 3D landmarks will be integrated into the environment map. Moreover, new 3D landmarks are created through feature matching and triangulation among the current keyframe and its associated keyframes in the local map. Afterwards, the 3D landmarks and camera poses in the local map of the current keyframe are refined by local bundle adjustment. In addition, the loop closure in the ORB-SLAM2 method is directly used.

During the creation of new 3D landmarks, the 3D coordinates of endpoints of each line segment are estimated according to reprojection constraints in three views. Similarly, for each 2D junction, 3D coordinates in its corresponding 3D line pair are calculated and the distance between these two coordinates is used to determine whether the 2D junction is a coplanar junction. On this basis, triangulation is performed to create new 3D junctions. Specially, each junction is attached to a confidence of coplanarity for calculating its covariance matrix of the local bundle adjustment. Besides, for each 3D junction of the current keyframe from the tracking stage, its confidence of coplanarity can be dynamically updated.

IV. Point-Line-Junction Based SLAM

In this section, ORB points [7], line segments, and junctions of coplanar lines are combined to achieve a monocular visual SLAM. Next, feature extraction and matching, feature tracking, and local mapping are addressed, respectively.

A. Feature Extraction and Matching

For the line segment feature, LSD [37] is employed for fast extraction. The candidate line segments will be filtered by a given length threshold. Afterwards, LBD descriptor [38] is calculated to express each line segment for line matching. Let $LineSet_I = \{l_1, l_2, \cdots, l_n\}$ denotes the line segment set extracted from the image I, where n refers to the number of line segments. It is worth mentioning that the stability of a junction is affected by the length of extension of corresponding line segment. Inspired by this, adaptive coordinate confidence is designed to evaluate the junction. Different from existing junction selection methods with fixed distance [32], [36], we consider generating more candidate junctions. The confidence is illustrated in Fig. 2. Take a line segment l_i with the length of D_{li} as an example. Its two endpoints S_i and E_i are extended to points Sx_i and Ex_i with the length λD_{li}, respectively, where λ is a given constant. Then, we get the confidence interval Ψ_i, which is between Sx_i and Ex_i. Junctions beyond this interval are considered invalid. The coordinate confidence of a junction P_i relative to line segment

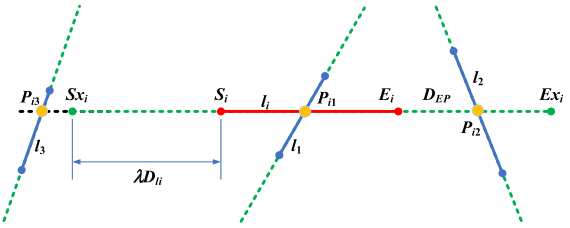


Fig. 2. Schematic diagram of junction coordinate confidence. The junction P_{i2} relative to line segment l_i has a confidence of $1 - \frac{D_{EP}}{\lambda D_{li}}$. The coordinate confidences of the junctions P_{i1} and P_{i3} relative to line segment l_i are 1 and 0, respectively.

l_i is calculated as follows.

$$Conf(P_i|l_i) = \begin{cases} 1 - \dfrac{D_{EP}}{\lambda D_{li}} & P_i \in \Psi_i \\ 0 & P_i \notin \Psi_i \end{cases} \tag{1}$$

where D_{EP} denotes the minimal distance from junction P_i to line segment l_i.

Since each junction P_{ij} is formed by two line segments l_i and l_j, the coordinate confidence of the junction P_{ij} is given by:

$$Confidence\left(P_{ij}\right) = Conf(P_i|l_i) * Conf(P_i|l_j) \tag{2}$$

In the process of extracting features of junctions, the relationship matrix $M = \begin{bmatrix} c_{11} & \cdots & c_{1n} \\ \vdots & \ddots & \vdots \\ c_{n1} & \cdots & c_{nn} \end{bmatrix}$ between candidate junctions and its constructed lines is maintained for junction matching, where $c_{ij}(i, j = 1, 2, \cdots, n)$ is the index of the junction formed by l_i and l_j. Note that only junctions whose coordinate confidences exceed a given threshold are valid, and $c_{ij} = -1$ for invalid junctions. As the junction is generated depending on intersection of a line pair, the matching of junctions between frames can be deduced with line matching and the relationship matrices.

Formally, we consider two frames F_1 and F_2 whose relationship matrices are labeled as M^{F_1} and M^{F_2}, respectively. In the frame F_1, two line segments $l_i^{F_1}$ and $l_j^{F_1}$ form a junction $P_{ij}^{F_1}$, correspondingly, a junction $P_{mn}^{F_2}$ attached to line segments $l_m^{F_2}$ and $l_n^{F_2}$ in the frame F_2 is defined. With line matching pairs $\langle l_i^{F_1}, l_m^{F_2} \rangle$ and $\langle l_j^{F_1}, l_n^{F_2} \rangle$ between F_1 and F_2, $\langle P_{ij}^{F_1}, P_{mn}^{F_2} \rangle$ is a junction matching pair when $c_{ij}^{F_1} \geq 0, c_{mn}^{F_2} \geq 0$.

B. Feature Tracking

After the features are extracted, they are associated between frames to estimate camera pose in the feature tracking stage. To continuously construct the association within features, the motion of camera should be estimated, which is performed incrementally on the local map. The 3D features in the last frame will be firstly used to match with the current frame based on the constant velocity model. If the feature correspondences are not enough to support the pose estimation, features of the last reference keyframe will be projected to the current frame for matching. For the projected line and the matched

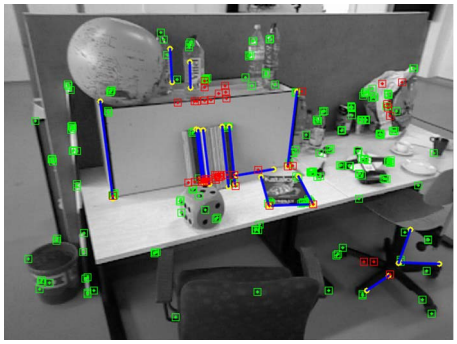


Fig. 3. Results of feature tracking for an image in sequence fr3_long_office, where the green points denote ORB features, the red points refer to the junctions of coplanar lines, and the blue line segments are line features.

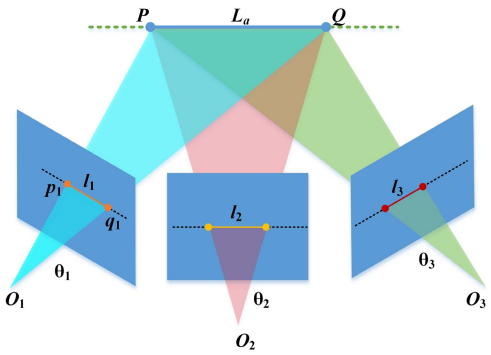


Fig. 4. 3D line segment creation within three views.

one, their overlap rate is considered to improve the accuracy of line matching. After the angles between the matched line and the coordinate axes in the image plane are calculated, the endpoints of the projected and the matched lines are projected on the coordinate axis with the smaller angle. On this coordinate axis, the overlap rate of line segments is calculated. The matching is considered as qualified when the overlap rate of line segments exceeds 85% [8]. Note that a junction is represented by a descriptor which combines the descriptors of its corresponding two line segments. In some sense, the junction can be regarded as an independent feature after creation. As long as the junction descriptors match normally, the junctions between frames can be matched even if the junction locations are occluded. For example, a junction is located at the extension lines of two line segments and there exists occlusion on the junction location. Because the occlusion does not affect the line segment descriptors, the junction descriptors are not affected and thus the matching of junctions between two frames is still normal.

After enough 3D-2D feature correspondences are obtained, an initial camera pose can be estimated through pose graph optimization. Afterwards, the local map of the current frame is constructed based on the co-visibility graph. The initial pose is then iteratively optimized by local bundle adjustment within the local map and the final camera pose is obtained. Next, it shall be judged whether the current frame is inserted into the map as a new keyframe, where the quantity variations of lines and junctions are also considered upon the existing criteria in ORB-SLAM2. Fig. 3 gives an illustration of feature tracking for an image in sequence fr3_long_office of the TUM RGB-D dataset, where the green points denote ORB features, the red points refer to the junctions of coplanar lines, and the blue line segments are line features. Some junctions seem to be isolated because they are located at the extension lines of line segments or their associated line segments are missing due to line tracking failure. Only the features that are successfully matched in the current frame are provided for clear presentation.

C. Local Mapping

Local mapping consists of keyframe preprocessing, new 3D line segment creation, new 3D junction creation and updating, and local bundle adjustment with multiple features. And new 3D landmarks are indispensable to enhance the quality of tracking, which are mainly created by triangulation among the feature correspondences between the current keyframe and its associated keyframes.

1) Keyframe Preprocessing: When a new keyframe arrives, its co-visibility graph is updated, where the shared line and junction features with other keyframes are also considered to enhance the co-visibility. Then, the culling involving ORB, line segment, and junction features is conducted to choose stable 3D features, which are inserted into the map. A feature is stable [6] when it is visible in at least 25% of the frames during the first several keyframes after creation and at least two keyframes can observe it.

2) New 3D Line Segment Creation: Since 2D detection may exist the shift problem of line segment endpoints [40], it is inaccurate to use endpoints information of 2D line segments between two keyframes for 3D line segment creation. Due to the fact that line correspondences between two views do not provide any constraints on the relative motion estimation [39], it is necessary to import a third keyframe during the creation of new 3D line segments to limit the line-based geometry within two views. The creation of 3D line segments in three views is shown in Fig. 4.

Formally, 3D line segment L_a is expressed by its 3D endpoints $P, Q \in \mathbb{R}^3$ and let l_k denote its 2D detections on image planes whose optical centers are located at O_k, where $k = 1, 2, 3$. We label $\boldsymbol{\theta}_k = \{\boldsymbol{R}_k, \boldsymbol{t}_k\} \in \mathbb{R}^{3\times4}$ as the camera pose, where $\boldsymbol{R}_k$ is the rotation matrix and $\boldsymbol{t}_k$ refers to the translation vector. The line coefficient vector $\mathbf{I}_k$ of the 2D line segment l_k is obtained as follows.

$$\mathbf{I}_k = p_k \times q_k \tag{3}$$

where p_k, q_k are 2D endpoints of l_k.

We label p_1 and q_1 as the endpoints of a 2D line segment l_1 in the current keyframe. To avoid the influence from the endpoint shift, in other two associated keyframes, we mainly concern the lines where 2D line segments related to l_1 are located regardless of their endpoints. Combined with the constraints that the projections of 3D line segment in other two keyframes are located on the corresponding 2D lines, we have:

$$\begin{cases} x_j = \varphi(X_j, \boldsymbol{\theta}_1, \mathbf{K}) \\ \mathbf{I}_2^T \varphi\left(X_j, \boldsymbol{\theta}_2, \mathbf{K}\right) = 0 \\ \mathbf{I}_3^T \varphi\left(X_j, \boldsymbol{\theta}_3, \mathbf{K}\right) = 0 \end{cases} \tag{4}$$

where $j = 1, 2$, and $\mathbf{K}$ denotes the camera calibration matrix, and x_1 and x_2 are the coordinates of p_1 and q_1, respectively. X_1 and X_2 indicate the 3D coordinates of P and Q. $\varphi(X_j, \boldsymbol{\theta}_k, \mathbf{K})$ reflects the projection from 3D to 2D, where $k = 1, 2, 3$. Then, the coordinates X_1 and X_2 of P and Q are solved by SVD decomposition.

$$\begin{bmatrix} \mathbf{I}_2^T \boldsymbol{T}_2 \\ \mathbf{I}_3^T \boldsymbol{T}_3 \\ x_{j,x}\boldsymbol{T}_1(3) - \boldsymbol{T}_1(1) \\ x_{j,y}\boldsymbol{T}_1(3) - \boldsymbol{T}_1(2) \end{bmatrix} X_j = \mathbf{0} \tag{5}$$

where $\boldsymbol{T}_k = \mathbf{K}\boldsymbol{\theta}_k \in \mathbb{R}^{3\times4}$, $\boldsymbol{T}_k(u)$ is the uth row of the matrix $\boldsymbol{T}_k$.

3) New 3D Junction Creation and Updating: With the 2D matched junctions from the current keyframe and its two associated keyframes, the 3D junction is created in three views as follows. If the 3D line segments L_a and L_b corresponding to the 2D line segments that form the 2D junction of the current keyframe are found, we combine the projected coordinates CP_{ab}^1 and CP_{ab}^2 of the 2D junction of the current keyframe on L_a and L_b to judge whether a 3D junction is created, as illustrated in Fig. 5(a). Coplanarity verification is executed and the distance between CP_{ab}^1 and CP_{ab}^2 is calculated by $distance\left(CP_{ab}^1, CP_{ab}^2\right) = \|CP_{ab}^1 - CP_{ab}^2\|_2$. When $distance(\cdot) \leq d_{th}$ is satisfied, a coplanar junction CP_{ab} is created by averaging CP_{ab}^1 and CP_{ab}^2, where d_{th} is a user-specified threshold. When 3D line segments L_a and L_b are not simultaneously detected, 3D junction may also exist. In this case, they are mined by junction matching in three views. As shown in Fig. 5(b), a candidate 3D junction CP_{ab} is firstly calculated based on the matched 2D junction pair between the current keyframe and another keyframe, and then the projection error e between the projection cp_{ab} of CP_{ab} and its correspondence in the third view is calculated. Once the square of this projection error is less than a given threshold, CP_{ab} passes the coplanarity verification and it is confirmed as a new 3D junction. The 3D coordinate of CP_{ab} is estimated with the following constraints of 3D-2D correspondences in three views.

$$cp_k = \varphi\left(CP_{ab}, \boldsymbol{\theta}_k, \mathbf{K}\right), \quad k = 1, 2, 3 \tag{6}$$

where $cp_k (k = 1, 2, 3)$ describe the 2D matched junctions in three views. On this basis, CP_{ab} is obtained with SVD decomposition.

$$\begin{bmatrix} cp_{1,x}\boldsymbol{T}_1(3) - \boldsymbol{T}_1(1) \\ cp_{1,y}\boldsymbol{T}_1(3) - \boldsymbol{T}_1(2) \\ cp_{2,x}\boldsymbol{T}_2(3) - \boldsymbol{T}_2(1) \\ cp_{2,y}\boldsymbol{T}_2(3) - \boldsymbol{T}_2(2) \\ cp_{3,x}\boldsymbol{T}_3(3) - \boldsymbol{T}_3(1) \\ cp_{3,y}\boldsymbol{T}_3(3) - \boldsymbol{T}_3(2) \end{bmatrix} CP_{ab} = \mathbf{0} \tag{7}$$

where $cp_{k,x}$, $cp_{k,y}$ denote the coordinate values of cp_k, and $\boldsymbol{T}_k = \mathbf{K}\boldsymbol{\theta}_k \in \mathbb{R}^{3\times4}$.

After obtaining coordinates of new 3D junctions, the confidences of coplanarity and observations are also updated, where the former is inherited from their 2D coordinate confidences and observations N_{obs} are all set to 3, which means that

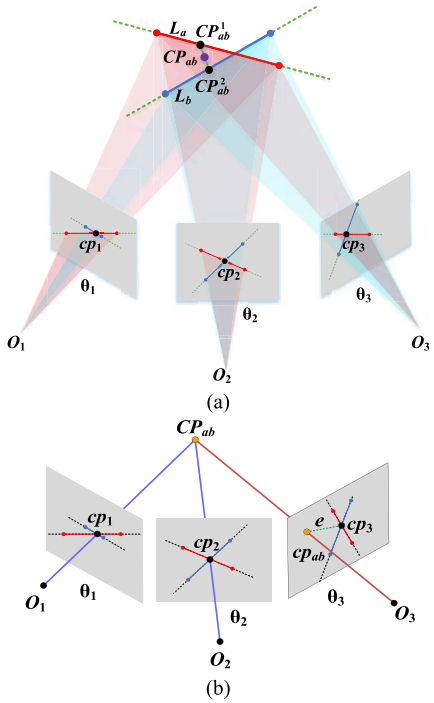


Fig. 5. 3D junction creation. (a) Coplanar junction creation based on 3D line correspondences. (b) Coplanar junction creation based on 2D junction correspondences.

reprojection error is satisfied in three views. The confidence of coplanarity is calculated as Equation (8) for local bundle adjustment.

$$Confidence_{3D}\left(CP_{ab}\right) = Confidence\left(cp_{ab}\right) \cdot \left(N_{obs} - 2\right) \tag{8}$$

For each 3D junction existing in the tracking stage, reprojection verification among keyframes will be performed to update its observation and confidence of coplanarity, where N_{obs} grows by 1 when the verification is passed. The detailed pipeline of 3D junction creation and updating is given in Algorithm 1, where P_{curr}^{2D} and P_{curr}^{3D} are 2D and 3D junction sets related to the current keyframe F_{curr}; $P_{curr,s}^{3D}$ denotes the 3D junction in P_{curr}^{3D} corresponding to a 2D junction p_s in P_{curr}^{2D}.

4) Local Bundle Adjustment With Multiple Features: The current camera pose and new landmarks will be further refined using local bundle adjustment. The keyframes and features including ORB, line segment, and junction of coplanar lines during optimization are divided into fixed and optimizable parts according to the closeness with the current keyframe. Specifically, all keyframes that have connection with the current keyframe in the co-visibility graph and features observed by these keyframes belong to the optimizable part, whereas keyframes that observe the aforementioned features but without connection to the current keyframe are categorized into the fixed part [6]. Next, the optimization errors of the line segments and the junction features are introduced.

For each 2D line segment l_k with line coefficient $\mathbf{I}_k = (l_k^1, l_k^2, l_k^3)$, the optimization errors related to endpoints P, Q (see Fig. 4) of 3D line segment L_a can be expressed by $e_p = \frac{\mathbf{I}_k^T \cdot \varphi(P, \boldsymbol{\theta}_k, \mathbf{K})}{\sqrt{(l_k^1)^2 + (l_k^2)^2}}$ and $e_q = \frac{\mathbf{I}_k^T \cdot \varphi(Q, \boldsymbol{\theta}_k, \mathbf{K})}{\sqrt{(l_k^1)^2 + (l_k^2)^2}}$ [39].

Algorithm 1 The Pipeline of 3D Junction Creation and Updating

Input: the current keyframe F_{curr}, its associated keyframes $\{\Phi_F\}$ and their camera poses, 2D and 3D junction sets $P^{2D}_{curr}, P^{3D}_{curr}$ related to F_{curr}
Output: the updated P^{3D}_{curr}
for p_s in P^{2D}_{curr} **do**
  if $P^{3D}_{curr,s} = NULL$ **do**
    Select F_u, F_v in $\{\Phi_F\}$ by co-visibility graph of F_{curr};
    if there exist correspondences of p_s in F_u and F_v **do**
      if a coplanar junction needs to be created **do**
        Obtain CP_{ab};
        $P^{3D}_{curr} \leftarrow CP_{ab}(N_{obs} = 3, Confidence_{3D}(CP_{ab}))$;
      end if
    end if
  else
    perform reprojection verification;
    if verification is passed **do**
      $N_{obs}++$;
      $P^{3D}_{curr,s} \leftarrow CP_{ab}(N_{obs}, Confidence_{3D}(CP_{ab}))$;
    end if
  end if
end for
return

The optimization of junction CP_{ab} with its 2D projection cp_{ab} is conducted by minimizing the projection error, and the error e_{junc} is given by:

$$e_{junc} = cp_{ab} - \varphi\left(CP_{ab}, \boldsymbol{\theta}_k, \mathbf{K}\right) \tag{9}$$

Then, a unified cost function to integrate optimization errors of all 3D-2D correspondences is given by:

$$\begin{aligned} cost = \sum \Big(& \rho_{orb} e_{orb}^{\mathrm{T}} \Omega_{orb}^{-1} e_{orb} \\ & + \rho_{line} \left(e_p^{\mathrm{T}} \Omega_{line}^{-1} e_p + e_q^{\mathrm{T}} \Omega_{line}^{-1} e_q \right) \\ & + \rho_{junc} e_{junc}^{\mathrm{T}} \Omega_{junc}^{-1} e_{junc} \Big) \end{aligned} \tag{10}$$

where e_{orb} is the reprojection error of an ORB point; $\rho_{orb}, \rho_{line}, \rho_{junc}$ are the huber cost functions for ORB, line, and junction features, respectively. Ω_{orb}, Ω_{line}, Ω_{junc} refer to the covariance matrices associated with ORB points, line endpoints, and junctions, respectively. And $\Omega_{orb} = \sigma^2_{orb}\mathbf{I}_{2\times 2}, \Omega_{line} = \sigma^2_{line}\mathbf{I}_{2\times 2}$, and $\Omega_{junc} = Confidence_{3D}(\cdot)\,\sigma^2_{junc}\mathbf{I}_{2\times 2}$, where σ^2_{orb}, σ^2_{line}, σ^2_{junc} are given variances.

V. Experimental Results

In this section, we evaluate the proposed monocular PLJ-SLAM method on the TUM RGB-D dataset [41]. Also, the extension of PLJ-SLAM with IMU is also tested on the EuRoC dataset [42]. The proposed method runs in an Intel Core i5-1135G7 CPU (4 cores @2.40GHz) with 16Gb RAM. We adopt absolute keyframe trajectory error (ATE) with root mean square as the evaluation metric, where ATE quantifies the difference between the estimated trajectory and ground truth. In the experiments, λ is set to 1.5 and d_{th} is equal to 2.5.

A. Ablation Study

To testify the performance of our proposed PLJ-SLAM method, its three variants PLJ-SLAM-I, PLJ-SLAM-II, and PLJ-SLAM-III are considered according to whether ORB, line, and junction are involved. Notice that the initialization parameter of ORB feature is 2000 in ORB-SLAM2. The comparison results of different methods on the sequences of the TUM RGB-D dataset are provided in Table I, where the best and the second-best results are labeled in red and blue, respectively. Compared to PLJ-SLAM-I with only ORB feature, PLJ-SLAM-II and PLJ-SLAM-III with the same ORB parameter achieve the improvement of pose estimation with the help of line feature or coplanar junction. In several sequences such as fr3_walk_xyz, there exists degradation of the performance. It could be because dynamic interference or small parallax affects the stability of three-view feature matching. To further improve the performance of pose estimation, the combination of ORB, line and junction features is helpful, which can be seen from our PLJ-SLAM. Besides, from the results of PLJ-SLAM-I, one can observe that there is performance improvement when the value of ORB initialization parameter increases, which means that the method with only ORB feature needs enough ORB points for the accuracy. On the contrary, PLJ-SLAM method does not need so many ORB points as other line and junction features can offer benefits. The whole performance is actually affected by multiple factors including feature types and environment. For PLJ-SLAM, its accuracy in 7 sequences is increased when setting the ORB initialization parameter from 2000 to 1200. When this parameter value is further limited (e.g. 800), performance shall become unstable due to the significant decreasing of features number. Compared with the results of PLJ-SLAM-II and PLJ-SLAM-1200, the introduction of the coplanar junction feature improves the accuracy. In general, the proposed PLJ-SLAM-1200 combines fewer ORB points with line and coplanar junction features to attain the best performance.

B. Comparison With Existing Methods

In this section, the proposed method is compared with existing methods including ORB-SLAM [6], PTAM [5], LSD-SLAM [10], LF-SLAM [43], PL-SLAM [8], and RGBD PLSLAM [13] on TUM RGB-D benchmark. The first three methods belong to point-based SLAM, the fourth method is line-based SLAM, and the last two methods correspond to the SLAM with point and line features. Since the monocular SLAM lacks scale information, the trajectories are first aligned in 7DoF with the ground truth. The comparison results on TUM RGB-D dataset are presented in Table II, where the best and the second-best results are labeled in red and blue, respectively. It can be seen from Table II that PTAM and LSD-SLAM fail in several sequences. Overall, our PLJ-SLAM achieves the best performance in 7 of 11 sequences, which demonstrates the effectiveness of the proposed method with points, lines, and junctions of coplanar lines. In the sequences fr2_xyz, fr1_floor, and fr2_360_kidnap, our PLJ-SLAM is possibly restricted by insufficient 3D junctions. For example, many long lines extracted from the sequence fr1_floor are almost in the same direction, which makes it difficult to form junctions.

TABLE I
COMPARISON OF DIFFERENT VARIANTS OF OUR PLJ-SLAM ON TUM RGB-D DATASET IN TERMS OF ATE (cm)

Method	ORB	Line	Junction	fr1_xyz	fr2_xyz	fr1_floor	fr2_360_kidnap	fr3_long_office	fr3_str_tex_far	fr3_str_tex_near	fr3_sit_xyz	fr3_sit_halfsph	fr3_walk_xyz	fr3_walk_halfsph
PLJ-SLAM-I	√ (1200)	×	×	1.22	0.32	X	5.28	2.91	1.16	1.51	X	1.55	X	2.18
	√ (2000)	×	×	1.14	0.30	2.15	5.03	1.98	1.14	1.43	1.15	1.45	1.25	1.84
PLJ-SLAM-II	√ (1200)	√	×	0.98	0.30	1.92	4.20	1.34	0.93	1.16	0.98	1.53	1.27	1.77
	√ (2000)	√	×	0.94	0.25	1.83	3.92	1.31	0.90	1.22	1.01	1.78	1.30	2.13
PLJ-SLAM-III	√ (1200)	×	√	0.96	0.30	1.67	3.79	2.10	1.13	1.18	1.04	1.44	1.45	2.00
	√ (2000)	×	√	1.01	0.26	1.83	2.97	2.74	1.04	1.07	1.05	1.46	1.53	1.85
PLJ-SLAM	√ (1200)	√	√	0.80	0.27	1.81	3.36	1.23	0.85	1.00	1.01	1.28	1.01	1.58
	√ (2000)	√	√	0.80	0.23	1.99	3.78	1.21	1.07	1.08	0.87	1.43	1.41	1.73
	√ (800)	√	√	0.92	0.36	8.94	3.40	1.55	0.89	1.35	0.94	1.70	X	X

The symbol "X" means the tracking failure. The best and second-best results are labeled in red and blue, respectively.

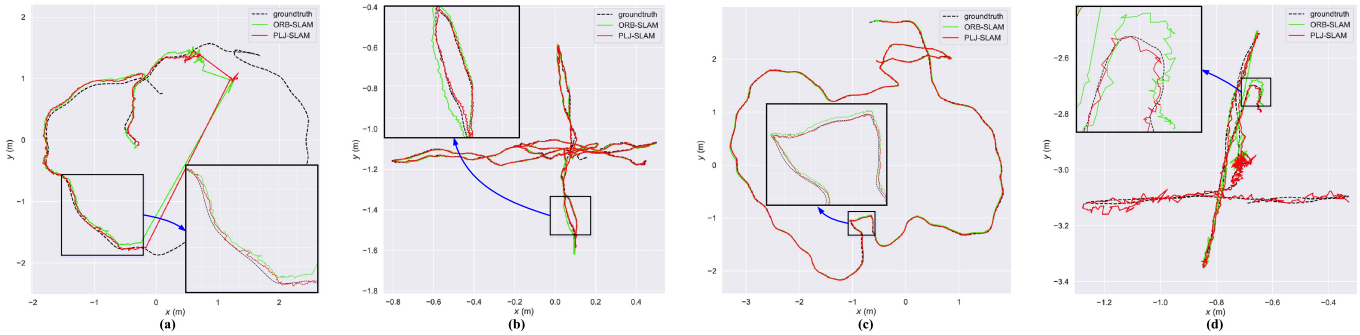


Fig. 6. Comparison of trajectories estimated by PLJ-SLAM and ORB-SLAM on four sequences of the TUM RGB-D dataset. (a) fr2_360_kidnap. (b) fr2_xyz. (c) fr3_long_office. (d) fr3_walk_xyz.

TABLE II
COMPARISON WITH EXISTING METHODS ON TUM RGB-D DATASET IN TERMS OF ATE (cm)

TUM RGB-D Sequences	ORB-SLAM [6]	PTAM [5]	LSD-SLAM [10]	LF-SLAM [43]	PL-SLAM [8]	RGBD PLSLAM [13]	PLJ-SLAM
fr1_xyz	0.90	1.15	9.00	1.05	1.21	1.16	0.80
fr2_xyz	0.30	0.20	2.15	0.25	0.43	0.40	0.27
fr1_floor	2.99	X	38.07	1.74	7.59	6.22	1.81
fr2_360_kidnap	3.81	2.63	X	2.97	3.92	3.26	3.36
fr3_long_office	3.45	X	38.53	1.35	1.97	1.86	1.23
fr3_str_tex_far	0.77	0.93	7.95	0.88	0.89	0.92	0.85
fr3_str_tex_near	1.58	1.04	X	1.17	1.25	1.03	1.00
fr3_sit_xyz	0.79	0.83	7.73	-	0.066	0.07	1.01
fr3_sit_halfsph	1.34	X	5.87	1.29	1.31	-	1.28
fr3_walk_xyz	1.24	X	12.44	1.16	1.54	1.60	1.01
fr3_walk_halfsph	1.74	X	X	1.66	1.60	1.58	1.58
Average	1.72	1.13*	15.22*	1.35*	1.98	1.81*	1.29

The data of existing methods were extracted from [6], [8], [13], and [43]. "*" reflects the average result of partial data and "-" means that the data is not provided in their papers.

Fig. 6 shows the comparison of trajectories estimated by PLJ-SLAM and ORB-SLAM on four sequences of TUM RGB-D dataset. It is noted that the camera's field of view was completely blocked manually in the middle part of sequence fr2_360_kidnap, and naturally the visual-based camera pose estimation fails. At the final stage of this sequence, the occlusion disappears and the camera pose estimation is restored by relocation. One can see that our trajectories are closer to the ground truth than those of ORB-SLAM.

C. Efficiency Analysis

In our method, there are three threads: tracking, local mapping, and loop closure. The first one is used to extract features and estimate camera pose based on feature matching, and the second one performs map maintenance, which includes keyframe preprocessing, new landmark creation, and local bundle adjustment. Note that the third thread is directly borrowed from ORB-SLAM2, which is not our focus. Table III presents the running time comparison of tracking and local mapping threads of PLJ-SLAM-1200, PL-SLAM and ORB-SLAM on the TUM RGB-D dataset. For the proposed method, the running time of these two main threads averaged on the sequences of the TUM RGB-D dataset are 87.7ms and 559.6ms, respectively. Besides ORB points, the proposed method still needs to extract line segments and junctions of coplanar lines. Thus, its running time is longer than that of

TABLE III
COMPARISON OF THE TRACKING AND LOCAL MAPPING THREADS OF DIFFERENT METHODS ON TUM RGB-D DATASET IN TERMS OF RUNNING TIME (ms)

TUM RGB-D sequences	PLJ-SLAM		PL-SLAM		ORB-SLAM	
	Tracking thread	Local mapping thread	Tracking thread	Local mapping thread	Tracking thread	Local mapping thread
fr1_xyz	90.5	652.5	91.8	861.8	50.5	211.1
fr2_xyz	74.1	367.7	113.8	982.8	51.9	235.4
fr1_floor	80.2	443.3	75.1	823.6	47.9	177.4
fr2_360_kidnap	54.8	239.8	72.4	561.2	40.0	138.8
fr3_long_office	97.5	692.2	111.9	1002.4	55.7	236.6
fr3_str_tex_far	98.2	865.3	112.9	1256.7	49.4	200.8
fr3_str_tex_near	94.1	850.8	104.8	1045.3	51.6	210.1
fr3_sit_xyz	97.3	582.5	111.5	830.5	46.8	153.4
fr3_sit_halfsph	91.6	472.5	109.2	776.4	47.1	170.4
fr3_walk_xyz	95.6	483.3	86.5	574.6	45.1	129.2
fr3_walk_halfsph	90.7	506.1	102.7	705	41.1	140.7
Average	87.7	559.6	99.3	856.4	47.9	182.2

ORB-SLAM. Compared to PL-SLAM, although the proposed method additionally adds the junctions, the junction feature in turn moderately reduces the possibility of other features involving pose optimization, and the average running time of our method is shorter. As the tracking thread outputs the pose of visual odometry without waiting the optimized map from the local mapping thread, the pose output frequency of the system is decided by the tracking thread regardless of other threads. In addition, the local mapping thread in our method mainly maintains the local map near the current keyframe, and its processing time is irrelevant to the scale of the environment. In general, the proposed method meets the requirement of normal tasks.

D. Robustness Verification

To further verify the proposed method, the disturbance is imposed on images, where the blur and brightness of images are artificially changed. Take a segment of sequence fr3_long_office as an example (see the green box in Fig. 7), and the results are depicted in Fig. 7. The Gaussian blurs are introduced where kernel sizes are 9×9 and 11×11 with standard deviations of $\sigma_x = \sigma_y = 3$ and $\sigma_x = \sigma_y = 4$, respectively. Fig. 7(a1) provides the trajectories of camera pose on the XOY plane, and Figs. 7(b1) and (c1) reflect the variations of translation and rotation components, respectively. It is seen that the proposed method can deal with the polluted images and the results are still close to the ground truth. The results of brightness interference by adding an increment to all pixels of an image can be found in Figs. 7(a2), (b2), and (c2), where the brightness increments are set to 50 and 100. In spite of perturbations, the outputs of our method are feasible. Figs. 8(b)-(d) present an illustration on an image (see Fig. 8(a)) from sequence fr3_long_office, which proves that our method can handle external interference.

E. Extension of PLJ-SLAM With IMU

With the development of visual SLAM, some methods have combined other sensors such as inertial measurement unit (IMU) for more robust pose estimation. In this section, we refer to [44] to incorporate IMU. An IMU with respect to the body frame B is used to measure the rotation rate and acceleration, which are affected by sensor biases b_g, b_a in addition to noises. Next, we use the IMU preintegration scheme, where position p and velocity v are described by the relative motion increments ΔR, Δv, and Δp between two consecutive keyframes as follows [44]:

$$\begin{cases} R_{WB}(k+1) = R_{WB}(k)\,\Delta R_{k,k+1} Exp\left(J^g_{\Delta R} b^k_g\right) \\ v(k+1) = v(k) + g_W \Delta t_{k,k+1} \\ \qquad + R_{WB}(k)\left(\Delta v_{k,k+1} + J^g_{\Delta v} b^k_g + J^a_{\Delta v} b^k_a\right) \\ p(k+1) = p(k) + v(k)\Delta t_{k,k+1} + \frac{1}{2} g_W \Delta t^2_{k,k+1} \\ \qquad + R_{WB}(k)\left(\Delta p_{k,k+1} + J^g_{\Delta p} b^k_g + J^a_{\Delta p} b^k_a\right) \end{cases} \tag{11}$$

where g_W refers to the gravity vector, and R_{WB} denotes the rotation from body frame B to the inertial frame W. Jacobians $J^g_{(\cdot)}$ and $J^a_{(\cdot)}$ denote the first-order approximation of corresponding biases [44]. The aforementioned IMU scheme is incorporated into our PLJ-SLAM, which is termed as PLJ-VI-SLAM.

TABLE IV
COMPARISON OF DIFFERENT METHODS ON THE EUROC DATASET IN TERMS OF ATE (m)

Methods	MH01	MH02	MH03	MH04	MH05
OKVIS	0.160	0.220	0.240	0.340	0.470
ROVIO	0.210	0.250	0.250	0.490	0.520
VINS-Mono	0.084	0.105	0.074	0.122	0.147
VI-DSO	0.062	0.044	0.117	0.132	0.121
ORB-SLAM3	0.062	0.037	0.046	0.075	0.057
PLJ-VI-SLAM	0.053	0.034	0.051	0.121	0.075

The best and second-best results are labeled in red and blue, respectively.

The comparison of PLJ-VI-SLAM with other monocular visual-inertial methods including OKVIS [45], ROVIO [46], VINS-Mono [47], VI-DSO [48], and ORB-SLAM3 [49] on the EuRoC dataset is illustrated in Table IV. One can see that ORB-SLAM3 and our PLJ-VI-SLAM perform well. Compared with ORB-SLAM3, our multi-feature combination scheme obtains better results in the bright sequences MH01 and MH02 with good texture. In the bright scene MH03 with fast motion, our method is close to the result of ORB-SLAM3. In addition to fast motion, the sequences

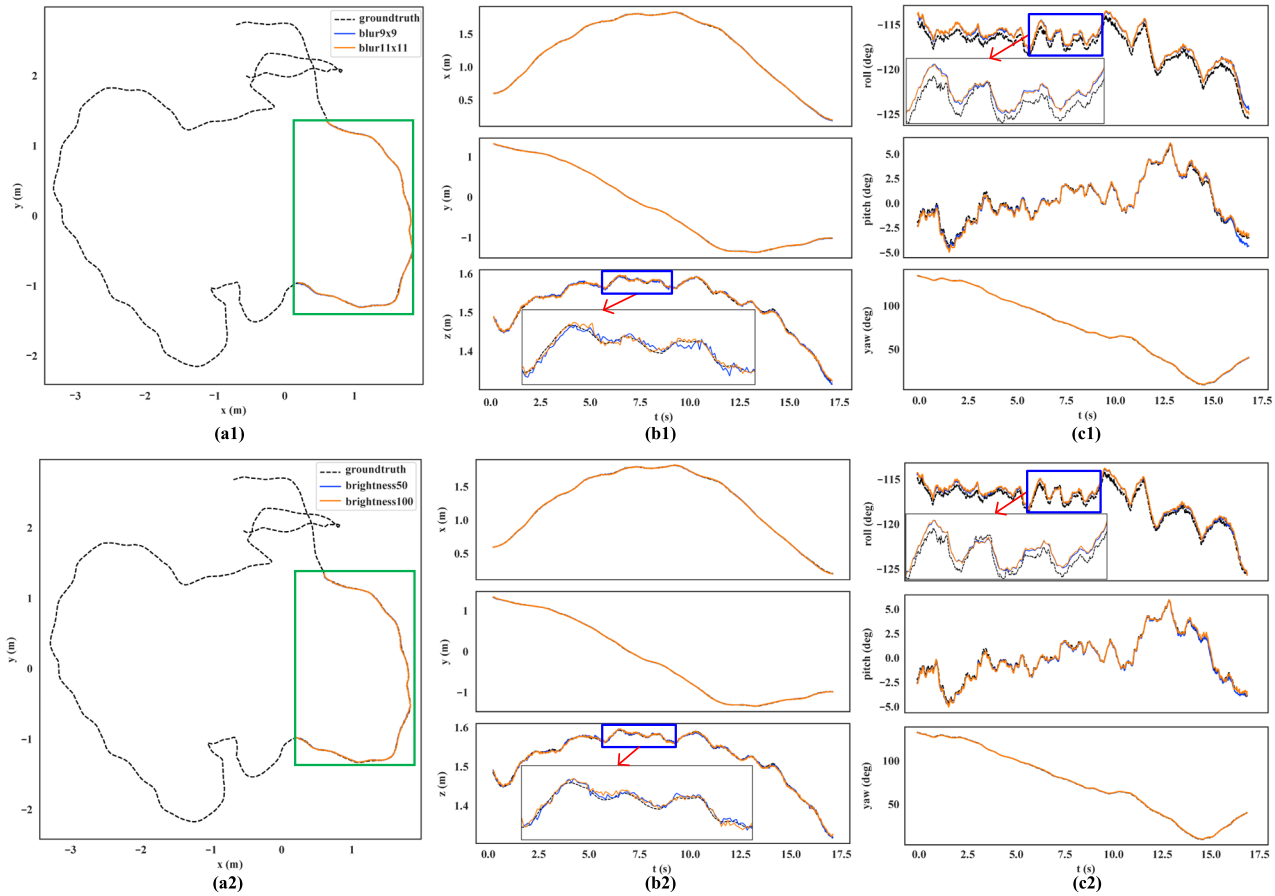
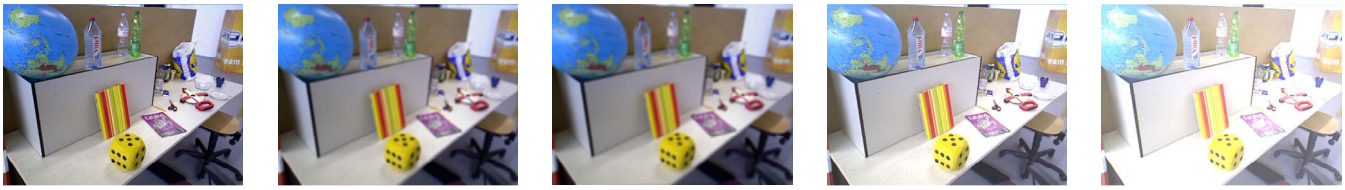


Fig. 7. Robustness verification with varying blur and brightness on a segment of fr3_long_office. (a1), (a2) The trajectories of camera pose on the XOY plane with different blur degrees and brightness increments. (b1), (c1) The variations of translation and rotation components, with Gaussian blurs whose kernel sizes are 9 × 9 and 11 × 11. (b2), (c2) The variations of translation and rotation components, with different brightness increments of 50 and 100.



(0.604m, 1.350m, 1.489m, -114.50°, -2.01°, 135.44°) (a)
(0.607m, 1.344m, 1.489m, -113.68°, -2.45°, 135.24°) (b)
(0.608m, 1.350m, 1.493m, -113.65°, -2.49°, 135.39°) (c)
(0.612m, 1.351m, 1.495m, -114.07°, -2.39°, 135.07°) (d)
(0.606m, 1.355m, 1.494m, -113.87°, -2.47°, 135.16°) (e)

Fig. 8. The results of an image from sequence fr3_long_office with blur and brightness interferences. (a) The original image with ground truth. (b), (c) The images with Gaussian blurs whose kernel sizes are 9 × 9 and 11 × 11, respectively. (d), (e) The images with different brightness increments of 50 and 100.

MH04 and MH05 exist both bright and dark fragments, which disturbs the matching of feature descriptors and thus decreases the performance of our method. Moreover, the introduction of multi-map data association technology in ORB-SLAM3 also further improves its pose estimation accuracy. From the whole results, the proposed method is considered as an effective one.

VI. Conclusion

In this paper, we present a multi-feature monocular visual SLAM system using points, lines, and junctions of coplanar lines. By exploring the coplanarity of line pairs, we propose a creation algorithm to obtain 3D junctions of coplanar lines from matching pairs of 2D junctions. Moreover, a unified optimization model is constructed to concurrently minimize reprojection errors of points, lines, and junctions of coplanar lines in the bundle adjustment. The experimental results verify the effectiveness of our method. In the future work, we shall combine higher-order constraints among the lines to further improve the performance of SLAM, including the inter-frame consistency of feature combination relation during the tracking stage. Besides, multiple visual features shall be integrated with more information including semantics to achieve a more accurate SLAM.

REFERENCES

[1] T. Shan and B. Englot, "LeGO-LOAM: Lightweight and ground-optimized lidar odometry and mapping on variable terrain," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2018, pp. 4758–4765.

[2] J. Zhang and S. Singh, "LOAM: Lidar odometry and mapping in real-time," in *Robotics: Science and Systems*. 2014.

[3] C.-H. Chien, C.-C.-J. Hsu, W.-Y. Wang, and H.-H. Chiang, "Indirect visual simultaneous localization and mapping based on linear models," *IEEE Sensors J.*, vol. 20, no. 5, pp. 2738–2747, Mar. 2020.

[4] 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.

[5] G. Klein and D. Murray, "Parallel tracking and mapping for small AR workspaces," in *Proc. 6th IEEE ACM Int. Symp. Mixed Augmented Reality*, Nov. 2007, pp. 225–234.

[6] 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.

[7] 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.

[8] A. Pumarola, A. Vakhitov, A. Agudo, A. Sanfeliu, and F. Moreno-Noguer, "PL-SLAM: Real-time monocular visual SLAM with points and lines," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2017, pp. 4503–4508.

[9] R. A. Newcombe, S. J. Lovegrove, and A. J. Davison, "DTAM: Dense tracking and mapping in real-time," in *Proc. Int. Conf. Comput. Vis.*, Nov. 2011, pp. 2320–2327.

[10] J. Engel, T. Schöps, and D. Cremers, "LSD-SLAM: Large-scale direct monocular SLAM," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 834–849.

[11] J. Engel, V. Koltun, and D. Cremers, "Direct sparse odometry," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 3, pp. 611–625, Mar. 2016.

[12] K. Qian, W. Zhao, K. Li, X. Ma, and H. Yu, "Visual SLAM with BoPLW pairs using egocentric stereo camera for wearable-assisted substation inspection," *IEEE Sensors J.*, vol. 20, no. 3, pp. 1630–1641, Feb. 2020.

[13] Q. Fu *et al.*, "A robust RGB-D SLAM system with points and lines for low texture indoor environments," *IEEE Sensors J.*, vol. 19, no. 21, pp. 9908–9920, Nov. 2019.

[14] Y. Salaün, R. Marlet, and P. Monasse, "Robust and accurate line-and/or point-based pose estimation without Manhattan assumptions," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 801–818.

[15] Y. Liu, D. Yang, J. Li, Y. Gu, J. Pi, and X. Zhang, "Stereo visual-inertial SLAM with points and lines," *IEEE Access*, vol. 6, pp. 69381–69392, 2018.

[16] R. Wang, K. Di, W. Wan, and Y. Wang, "Improved point-line feature based visual SLAM method for indoor scenes," *Sensors*, vol. 18, no. 10, p. 3559, Oct. 2018.

[17] S. J. Lee and S. S. Hwang, "Elaborate monocular point and line SLAM with robust initialization," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 1121–1129.

[18] H. Kim and S. Lee, "A novel line matching method based on intersection context," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2010, pp. 1014–1021.

[19] D. C. Herrera, K. Kim, J. Kannala, K. Pulli, and J. Heikkila, "DT-SLAM: Deferred triangulation for robust SLAM," in *Proc. 2nd Int. Conf. 3D Vis.*, Dec. 2014, pp. 609–616.

[20] S. Maity, A. Saha, and B. Bhowmick, "Edge SLAM: Edge points based monocular visual SLAM," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops (ICCVW)*, Oct. 2017, pp. 2408–2417.

[21] E. Rosten and T. Drummond, "Machine learning for high-speed corner detection," in *Proc. Eur. Conf. Comput. Vis.*, 2006, pp. 430–443.

[22] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: An efficient alternative to SIFT or SURF," in *Proc. Int. Conf. Comput. Vis.*, Nov. 2011, pp. 2564–2571.

[23] J. Li *et al.*, "Attention-SLAM: A visual monocular SLAM learning from human gaze," *IEEE Sensors J.*, vol. 21, no. 5, pp. 6408–6420, Mar. 2021.

[24] P. Smith, I. Reid, and A. J. Davison, "Real-time monocular SLAM with straight lines," in *Proc. Brit. Mach. Vis. Conf.*, 2006, pp. 1–10.

[25] G. Zhang and I. H. Suh, "A vertical and floor line-based monocular SLAM system for corridor environments," *Int. J. Control, Autom. Syst.*, vol. 10, pp. 547–557, Jun. 2012.

[26] H. Zhou, D. Zou, L. Pei, R. Ying, P. Liu, and W. Yu, "StructSLAM: Visual SLAM with building structure lines," *IEEE Trans. Veh. Technol.*, vol. 64, no. 4, pp. 1364–1375, Apr. 2015.

[27] G. Zhang, D. H. Kang, and I. H. Suh, "Loop closure through vanishing points in a line-based monocular SLAM," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2012, pp. 4565–4570.

[28] J. Han Lee, G. Zhang, J. Lim, and I. H. Suh, "Place recognition using straight lines for vision-based SLAM," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2013, pp. 3799–3806.

[29] W. Jeong and K. Lee, "Visual SLAM with line and corner features," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Oct. 2006, pp. 2570–2575.

[30] R. Dong, F. Vincent, L. Simon, F. Isabelle, and C. Liu, "Line-based monocular graph SLAM," in *Proc. IEEE Int. Conf. Multisensor Fusion Integr. Intell. Syst. (MFI)*, Nov. 2017, pp. 494–500.

[31] H. Zhou, H. Fan, K. Peng, W. Fan, D. Zhou, and Y. Liu, "Monocular visual odometry initialization with points and line segments," *IEEE Access*, vol. 7, pp. 73120–73130, 2019.

[32] H. Kim and S. Lee, "Simultaneous line matching and epipolar geometry estimation based on the intersection context of coplanar line pairs," *Pattern Recognit. Lett.*, vol. 33, no. 10, pp. 1349–1363, 2012.

[33] K. Li *et al.*, "Hierarchical line matching based on line–junction–line structure descriptor and local homography estimation," *Neurocomputing*, vol. 184, pp. 207–220, Apr. 2016.

[34] E. Vincent and R. Laganière, "Junction matching and fundamental matrix recovery in widely separated views," in *Proc. Brit. Mach. Vis. Conf.*, 2004, pp. 1–10.

[35] H. Kim, S. Lee, and Y. Lee, "Wide-baseline stereo matching based on the line intersection context for real-time workspace modeling," *J. Opt. Soc. Amer. A, Opt. Image Sci.*, vol. 31, no. 2, pp. 421–435, 2014.

[36] J. Ma, X. Wang, Y. He, X. Mei, and J. Zhao, "Line-based stereo SLAM by junction matching and vanishing point alignment," *IEEE Access*, vol. 7, pp. 181800–181811, 2019.

[37] R. G. von Gioi, J. Jakubowicz, J.-M. Morel, and G. Randall, "LSD: A line segment detector," *IPOL J.*, vol. 2, pp. 35–55, Mar. 2012.

[38] 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.

[39] G. Zhang, J. H. Lee, J. Lim, and I. H. Suh, "Building a 3-D line-based map using stereo SLAM," *IEEE Trans. Robot.*, vol. 31, no. 6, pp. 1364–1377, Dec. 2015.

[40] A. Vakhitov, J. Funke, and F. Moreno-Noguer, "Accurate and linear time pose estimation from points and lines," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 583–599.

[41] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A benchmark for the evaluation of RGB-D SLAM systems," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Oct. 2012, pp. 573–580.

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

[43] Q. Wang, Z. Yan, J. Wang, F. Xue, W. Ma, and H. Zha, "Line flow based SLAM," 2020, *arXiv:2009.09972*.

[44] R. Mur-Artal and J. D. Tardós, "Visual-inertial monocular SLAM with map reuse," *IEEE Robot. Automat. Lett.*, vol. 2, no. 2, pp. 796–803, Apr. 2017.

[45] 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.

[46] M. Bloesch, M. Burri, S. Omari, M. Hutter, and R. Siegwart, "Iterated extended Kalman filter based visual-inertial odometry using direct photometric feedback," *Int. J. Robot. Res.*, vol. 36, no. 10, pp. 1053–1072, 2017.

[47] 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.

[48] L. Von Stumberg, V. Usenko, and D. Cremers, "Direct sparse visual-inertial odometry using dynamic marginalization," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2018, pp. 2510–2517.

[49] C. Campos *et al.*, "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.

Guangli Ren received the B.S. degree in intelligent science and technology from Dalian Maritime University, Dalian, China, in 2015, and the M.S. degree in technology of computer application from Capital Normal University, Beijing, China, in 2018. He is currently pursuing the Ph.D. degree in control theory and control engineering with the Institute of Automation, Chinese Academy of Sciences, Beijing. His current research interests include visual SLAM and robotic manipulation.

Zhiqiang Cao (Senior Member, IEEE) received the B.S. degree in industrial automation and the M.S. degree in control theory and control engineering from the Shandong University of Technology, Jinan, China, in 1996 and 1999, respectively, and the Ph.D. degree in control theory and control engineering from the Institute of Automation, Chinese Academy of Sciences, Beijing, China, in 2002. He is currently a Professor with the State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences. His research interests include SLAM, navigation, and manipulation of service robot.

Xilong Liu received the B.S. degree from Beijing Jiaotong University, Beijing, China, in 2009, and the Ph.D. degree in control theory and control engineering from the Institute of Automation, Chinese Academy of Sciences, Beijing, in 2014. He is currently an Associate Professor with the Research Center of Precision Sensing and Control, Institute of Automation, Chinese Academy of Sciences. His current research interests include image processing, pattern recognition, and visual measurement.

Min Tan received the B.E. degree from Tsinghua University, Beijing, China, in 1986, and the Ph.D. degree in control theory and control engineering from the Institute of Automation, Chinese Academy of Sciences, Beijing, in 1990. He is currently a Professor with the State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences. His research interests include advanced robot control and biomimetic robot.

Junzhi Yu (Fellow, IEEE) received the B.E. degree in safety engineering and the M.E. degree in precision instruments and mechanology from the North University of China, Taiyuan, China, in 1998 and 2001, respectively, and the Ph.D. degree in control theory and control engineering from the Institute of Automation, Chinese Academy of Sciences, Beijing, China, in 2003. From 2004 to 2006, he was a Postdoctoral Research Fellow with the Center for Systems and Control, Peking University. He was an Associate Professor with the Institute of Automation, Chinese Academy of Sciences, in 2006, where he was a Full Professor in 2012. In 2018, he joined the College of Engineering, Peking University, as a Tenured Full Professor. His current research interests include intelligent robots, motion control, and intelligent mechatronic systems.