IT-HYFAO-VO: Interpretation Tree-Based VO With Hybrid Feature Association and Optimization

Qinxuan Sun, Jing Yuan, *Member, IEEE*, and Xuebo Zhang, *Senior Member, IEEE*

***Abstract*—For the visual odometry (VO) and simultaneous localization and mapping (SLAM) in indoor environments, high-level geometric features have attracted more and more attention in recent years. Unlike the generally used point features, the geometric features, such as planes and lines, encode more higher-level semantic information of the scene which is beneficial for various tasks of mobile robots. In this article, an RGB-D VO system with hybrid high-level geometric features is developed. An interpretation tree (IT)-based hybrid feature association framework is proposed, which turns the feature association into a multiple-hypothesis decision problem. The IT expansion method is elaborately designed. Specifically, an internode consistency is proposed for generation of hypotheses and a consistent transformation model (CTM) for each hypothesis is explicitly expressed and incrementally updated. When the IT is constructed, a closed-form solution to the feature association and the camera transformation can be obtained. Then, a hybrid feature joint optimization method is introduced to further refine the pose estimate and parameters of geometric features. During optimization, the planes and lines are appropriately parameterized and the uncertainties arising from feature extraction are derived and used to balance the contributions of two types of features in the cost function. Extensive experiments are executed on public datasets and the results demonstrate that the proposed method can achieve higher accuracy and stronger robustness.**

***Index Terms*—High-level geometric features, joint multifeature association, RGB-D visual odometry (VO).**

I. Introduction

VISUAL odometry (VO) and simultaneous localization and mapping (SLAM) are important tasks in robotics [1]–[3]. In VO and SLAM systems, point features have been widely used and remarkable performance is achieved [4]–[6]. However, in low-textured environments, point features are scarce or not well-distributed in the image, and in this case, point features are ineffective to localize the camera and represent the structure of the environment [7]–[9]. In recent years, high-level geometric features have been exploited in VO and SLAM [10]. They can be combined with point features to gain robustness to low-textured environments as well as varying illumination conditions [7], [11]. Man-made environments are often dominated by objects from which the high-level geometric features, such as planes and lines, are easy to be extracted [7], [12], [13]. Compared with point features, geometric features are less numerous [14] and less sensitive to illumination changes and position ambiguity [13]. Moreover, the geometric feature-based map is more semantically meaningful compared with the point feature-based map, such as the Manhattan planes and edges in a man-made environment, which convey the high-level semantic information about the general geometric structure of the scene and are beneficial to high-level tasks of robots [15]–[17].

Two issues are raised in VO using high-level geometric features in 3-D space: establishing feature correspondences across successive frames, and properly parameterizing the features in the optimization. The existing methods for association of geometric features (planes and lines) in 3-D space can be classified into two categories, i.e., nearest neighbor (NN) search-based methods [8], [16], [18] and RANSAC-based methods [12], [14], [19].

For the NN search-based association, the best match for each feature is selected via an NN search using a predefined distance measure. The distance measure for plane features is often defined by geometric constraints (e.g., angles between plane normals and difference of distances from the origin to the planes) and the projection overlapping between planes [8], [16], [20]. For line features, the NN search-based association is mostly accomplished using the similarity between the visual descriptors extracted from the image. The line band descriptor (LBD) [21] is the most commonly used one in VO and SLAM systems [7], [13]. The NN search-based methods can also be used for the association of multiple features. For instance, plane and line features were used together in [11] and they were associated using the NN search based on the aforementioned distance measures. Nevertheless, the two different features were treated separately in the association and the geometric constraints between them were not considered. As a result, it may occur that the associated geometric features cannot be aligned by a common transformation of the camera because of incorrect correspondences. During the pose estimation, the associated features are usually jointly

Manuscript received May 30, 2021; revised August 7, 2021; accepted August 12, 2021. Date of publication August 24, 2021; date of current version September 6, 2021. This work was supported in part by the Natural Science Foundation of China under Grant 62073178, in part by the Tianjin Science Fund for Distinguished Young Scholars under Grant 20JCJQJC00140, in part by the Fundamental Research Funds for the Central Universities, Nankai University, under Grant 63206026, in part by the Major Basic Research Projects of Natural Science Foundation of Shandong Province under Grant ZR2019ZD07, and in part by the Tianjin Natural Science Foundation under Grant 20JCYBJC01470. The Associate Editor coordinating the review process was Jochen Lang. *(Corresponding author: Jing Yuan.)*

The authors are with the College of Artificial Intelligence, Nankai University, Tianjin 300350, China, and also with the Tianjin Key Laboratory of Intelligent Robotics, Nankai University, Tianjin 300350, China (e-mail: nkyuanjing@gmail.com).

This article has supplementary downloadable material available at https://doi.org/10.1109/TIM.2021.3107058, provided by the authors.

Digital Object Identifier 10.1109/TIM.2021.3107058

TABLE I
COMPARISON BETWEEN DIFFERENT ASSOCIATION FRAMEWORKS FOR GEOMETRIC FEATURES IN 3-D SPACE

	NN search	RANSAC	our framework
Consistent under a common transformation	×	✓	✓
Closed-form solution	✓	×	✓
Selection of feature pairs to determine consistent transformations	–	Randomly sampled pairs	All the hypothesized pairs
Computation of consistent transformations	–	Re-computing in each iteration	Incrementally constrain the solution

optimized. Thus, if correctly associated pairs are not sufficient to dominate the optimization, a fatal error may occur which might cause a disastrous breakdown of the whole system.

For the RANSAC-based association, a transformation model is computed using a set of randomly sampled correspondences and this process is iterated until sufficient inliers are obtained. In [12] and [14], plane and point features were associated via a RANSAC framework in case that plane features were insufficient for pose estimation. In [19], RANSAC was applied to associate the lines extracted from images and estimate the camera poses, which were further refined by nonlinear optimization. For the RANSAC-based methods, the associated features are ensured to be consistent under a common transformation model and can be used in the subsequent optimization. Nevertheless, the RANSAC-based association has three major disadvantages. First, the RANSAC is an *iterative process*, of which the runtime and convergence are sensitive to thresholds and the ratio of inliers. Second, in each iteration, the feature pairs used to determine the transformation model are *randomly sampled*. Thus, the possibility remains that the final consensus set is compatible with an incorrect model [22], [23]. Third, the hypothesized transformation model for each iteration is *recomputed* using newly sampled pairs and the constraints provided in the previous iterations are discarded, causing the waste of computing resources and slow convergence speed.

To fully address the aforementioned issues, a novel framework is proposed in this article for the association of multiple high-level geometric features and the comparison with the existing frameworks is shown in Table I. Compared with the NN search-based methods, both the RANSAC-based framework and the proposed framework guarantee that the associated features are consistent under a common transformation of the camera. Compared with the RANSAC-based framework, all the possible hypotheses are properly structured in our framework and an optimal solution can be determined in *a closed form*. Furthermore, instead of recomputing the transformation model using the randomly sampled feature correspondences, the transformation model corresponding to each hypothesis is *incrementally updated* in our framework.

The association framework proposed in this article is based on the interpretation tree (IT). An IT structure is spanned by all possible solutions to the data association problem [24]. The feature correspondences can be found by a constraint-based search in the IT. In [25] and [26], the IT was used to associate the line segments in 2-D space. The constraints applied for the tree search were based on the relation table, which was proposed as a representation of geometric patterns of line segments. In [27], the location-independent constraints were applied and the location of the robot was estimated for each hypothesis once the location can be fully constrained. Then, the location-dependent constraints were applied to further reduce the search space of the IT. Another important application of ITs in data association is the joint compatibility branch and bound (JCBB) algorithm proposed in [28], which traverses the IT in search for the hypothesis with the largest number of jointly compatible feature pairings. The JCBB algorithm has been adopted by 2-D SLAM systems performed in an EKF framework [29]–[32]. However, the IT structure has never been used in the joint association of multiple geometric features in 3-D space. Furthermore, though all the hypotheses can be traversed in the aforementioned IT-based association methods, they cannot provide an explicit relation of the constrained subspace of robot poses to the spatial configuration of features, which is significantly beneficial to the active observation and navigation of a robot. In contrast, in our framework, the problems of feature association and pose estimation are solved simultaneously in an incremental manner, and the constrained subspace of robot poses corresponding to each associated pair is explicitly represented.

The parameterization is another fundamental problem in the VO and SLAM systems using geometric features. In general, unlike point features, no dominant method is available for parameterization of high-level geometric features in 3-D space. The homogeneous coordinates were utilized to parameterize the plane in 3-D space [12], [23]. The Hessian normal form uses the unit normal and vertical distance from the origin to the plane to represent the plane [33], [34]. The closest point (CP) from the plane to the origin was used in [11], [35], and [36]. The unit quaternion parameterization with a minimal representation was employed in a factor-graph formulation of SLAM problems [37]. The minimal representation was used to update plane parameters during the factor-graph optimization. In this way, the parameters of planes can be estimated in the general graph-based SLAM systems [16], [20]. As for line features, the coordinates of end-points were usually adopted to represent a line [7], [10], [11], [38]. The Plücker coordinates [23] were also widely used for the geometrically simple representation of line transformations in 3-D space [33], [39], [40]. However, the Plücker coordinates are overparameterized which are inappropriate for the optimization of line parameters. Therefore, in the works of [13] and [41], an orthonormal representation proposed in [42] was used in the optimization process. The orthonormal representation is a minimal and decoupled representation and the conversion between the orthonormal representation and the Plücker coordinates is quite simple.

In this article, we develop an IT-based VO system with association and optimization of hybrid features (IT-HYFAO-VO). A novel framework is proposed based on the IT structure to associate multiple types of geometric features simultaneously.

All the possible pairs of features are organized in an IT structure, with each node representing a possible correspondence of features and an interpretation being a path from the root node to a leaf node, i.e., a set of feature correspondences. During the generation of interpretations, an internode consistency is defined between two nodes such that the feature correspondences represented by these two nodes can be aligned by common transformations of the camera. For any existing interpretation in the IT, any two nodes in the interpretation satisfy the inter-node consistency and the transformations that align the two nodes are explicitly expressed. Then, a consistent transformation model (CTM) is proposed for an interpretation, which consists of the transformations that can align the associated features represented by all the nodes in the interpretation. The CTM is incrementally updated as more nodes are added to the interpretation. After the association of geometric features and the calculation of the camera pose transformation, a hybrid feature joint optimization is introduced to further refine the camera pose and the parameters of the geometric landmarks. The original contributions of this article are summarized as follows.

1) An IT-based framework is proposed for the association of hybrid geometric features, which structures all the possible hypotheses with an IT and the optimal solution can be determined in a closed form. To the best of our knowledge, this is the first feature association algorithm to combine two types of different geometric features (planes and lines) in 3-D space into one unified framework. In addition, the proposed framework is theoretically extensible to other types of parameterized geometric features.
2) An incremental IT construction algorithm based on the internode consistency computation and CTM update is proposed. Specifically, through the computation of the internode consistency, the interpretations in the IT are generated. With the increase of each interpretation, the CTM is incrementally updated along with the construction of the IT, which can gradually constrain the feature association and pose estimation. After the construction of the IT structure, the closed-form solutions to the feature association as well as the camera pose estimation can be obtained simultaneously.
3) A scheme of hybrid feature joint optimization is proposed to refine the camera pose estimate as well as the parameters of the plane and line features, given the results of feature association and the initial estimate of the camera pose. The uncertainties arising from the extraction of the two types of features are computed to well balance the two different components in the joint optimization process. As a result, an accurate and robust VO system can be achieved.

The rest of this article is organized as follows. The system overview is presented in Section II. The IT-based hybrid geometric feature association is proposed in Section III. The hybrid feature optimization is presented in Section IV. Extensive experimental evaluations are presented in Section V. Conclusions are drawn in Section VI.

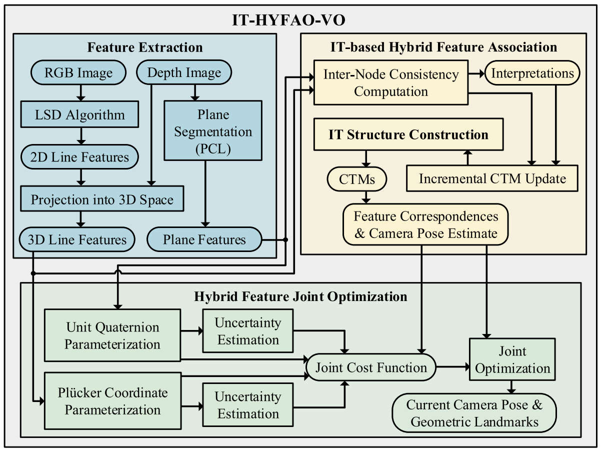

IT-HYFAO-VO
Feature Extraction
RGB Image
Depth Image
LSD Algorithm
Plane Segmentation (PCL)
2D Line Features
Projection into 3D Space
3D Line Features
Plane Features
IT-based Hybrid Feature Association
Inter-Node Consistency Computation
Interpretations
IT Structure Construction
CTMs
Incremental CTM Update
Feature Correspondences & Camera Pose Estimate
Hybrid Feature Joint Optimization
Unit Quaternion Parameterization
Uncertainty Estimation
Joint Cost Function
Joint Optimization
Plücker Coordinate Parameterization
Uncertainty Estimation
Current Camera Pose & Geometric Landmarks

Fig. 1. System overview.

II. SYSTEM OVERVIEW

The architecture overview of the IT-HYFAO-VO is shown in Fig. 1, which consists of three main components, i.e., feature extraction, IT-based hybrid feature association, and hybrid feature joint optimization.

The feature extraction module outputs the geometric features (planes and lines) extracted from RGB-D images. It should be noted that any plane/line extraction method can be used here as long as the 3-D planes/lines can be extracted in the camera coordinate system. In the implementation, the plane features are extracted from the depth image captured by an RGB-D camera, with the plane segmentation method [43] implemented in the Point Cloud Library (PCL). For the line feature extraction, the 2-D lines in the image are extracted from the RGB image by the LSD algorithm proposed in [44], which are projected into the 3-D space using the depth information to yield the 3-D line features. The plane and line features extracted from successive frames are fed into the IT-based hybrid feature association module and the hybrid feature joint optimization module, respectively. In the feature association module, the hypotheses about possible associations are organized in an IT structure. The interpretations in the IT are generated through the computation of the internode consistency to guarantee that any two nodes in an interpretation can be aligned by common transformations of the camera. For each generated interpretation, the CTM is incrementally updated. After construction of the IT, the resultant CTMs give all the associated feature pairs in an interpretation and the corresponding camera transformation, which are fed into the hybrid feature joint optimization module. In the hybrid feature joint optimization module, the plane and line features are properly parameterized for the nonlinear optimization and their uncertainties are estimated to well balance their contributions in the joint cost function. A more accurate camera pose estimate as well as a map composed of geometric landmarks are obtained through the joint optimization process.

III. IT-BASED HYBRID FEATURE ASSOCIATION

The IT-based hybrid feature association plays a central role in the IT-HYFAO-VO, which outputs feature correspondences

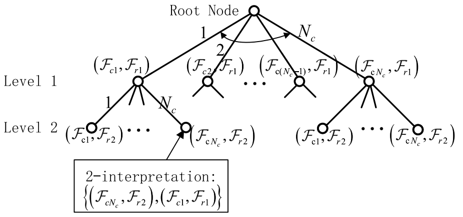

Root Node
1
2
N_c
$(\mathcal{F}_{c1},\mathcal{F}_{r1})$
$(\mathcal{F}_{c2},\mathcal{F}_{r1})$
$(\mathcal{F}_{c(N_c-1)},\mathcal{F}_{r1})$
$(\mathcal{F}_{cN_c},\mathcal{F}_{r1})$
Level 1
1
N_c
Level 2
$(\mathcal{F}_{c1},\mathcal{F}_{r2})$
$(\mathcal{F}_{cN_c},\mathcal{F}_{r2})$
$(\mathcal{F}_{c1},\mathcal{F}_{r2})$
$(\mathcal{F}_{cN_c},\mathcal{F}_{r2})$
2-interpretation:
$\{(\mathcal{F}_{cN_c},\mathcal{F}_{r2}),(\mathcal{F}_{c1},\mathcal{F}_{r1})\}$

Fig. 2. First two levels of a full IT structure.

and the estimated camera pose simultaneously. The hypotheses about possible associations of multiple geometric features are organized in an IT structure and the inherent geometric properties of features are fully exploited. We design a novel algorithm for the incremental construction of the tree. A new node is added to an interpretation if the internode consistency is satisfied, which is computed in Section III-A. For each interpretation, a CTM is incrementally updated to make sure that there exists a common transformation that aligns all the feature pairs. The CTM is introduced in Section III-B.

For the sake of clarity, we first define the notations used in this section. A geometric feature is represented by $\mathcal{F} \in \{\pi, \mathcal{L}\}$ which can be either a plane feature π or a line feature $\mathcal{L}$. $\pi = [\boldsymbol{n}^T, d]^T$ is the plane feature extracted from an RGB-D scan with $\boldsymbol{n} \in \mathbb{S}^2$ being the unit normal vector and $d \in \mathbb{R}$ the vertical distance from the origin to the plane. And $\mathcal{L} = [\boldsymbol{u}^T, \boldsymbol{v}^T]^T$ is the line feature, where $\boldsymbol{u} \in \mathbb{R}^3$ is a vector with its direction orthogonal to the plane defined by the join of the line and the origin, and its norm equal to the vertical distance from the origin to the line, and $\boldsymbol{v} \in \mathbb{S}^2$ is the unit direction vector of the line. The coordinate system in which the feature $\mathcal{F}$ is described as denoted by the subscript. In this section, we suppose that the IT is constructed for a frame-to-frame registration and the current and reference frames are denoted by the subscripts c and r, respectively.

The IT (Fig. 2) is a data structure that can be used to match features or geometrical primitives in two different coordinate systems. Each node $\mathcal{N} = (\mathcal{F}_c, \mathcal{F}_r)$ in the IT represents a correspondence between a feature $\mathcal{F}_c$ from the current frame and a feature $\mathcal{F}_r$ from the reference frame. An n-interpretation $\mathcal{P}_n = \{\mathcal{N}^n, \mathcal{N}^{n-1}, \ldots, \mathcal{N}^1\}$ is a path from the root node to a node at the nth level of the IT, which is a set of n pairings of features from two frames. Note that, $\mathcal{N}^j$, $j = 1, \ldots, n$ can be any node at the jth level. Because the nodes are handled within an interpretation through this section, we omit the indices that label different nodes at the same level for brevity. The constructed IT has N_r levels in total, where N_r is the number of features in the reference frame. All possible pairings of the jth $(j = 1, \ldots, N_r)$ feature in the reference frame with the features in the current frame are at the jth level of the tree. The branching factor at each node is N_c which is the number of features in the current frame, i.e., each node can have N_c descendants at most. Fig. 2 shows the first two levels of a full IT structure, which contains all the possible associations of the features observed in two different camera frames. It is clear that the full IT is highly redundant [45] and includes plenty of incorrect associations.

A. Internode Consistency

The internode consistency is proposed as a judgment to determine whether a new node $\mathcal{N}^{n+1}, n = 1, \ldots, N_r - 1$ can be added to an existing interpretation $\mathcal{P}_n$ in the IT. That is to say, if the internode consistency between $\mathcal{N}^{n+1}$ and any node $\mathcal{N}^i, i = 1, \ldots, n$ in $\mathcal{P}_n$ is satisfied, the node $\mathcal{N}^{n+1}$ will be added to $\mathcal{P}_n$ yielding $\mathcal{P}_{n+1}$. The internode consistency guarantees that for any two nodes that are in the same interpretation in the IT, the corresponding two pairs of features can be aligned by common transformations of the camera. Together with the CTM presented in the next subsection, the association of features and the localization of the camera can be achieved directly after the IT is constructed.

Definition 1 (Internode Consistency): Given two nodes in an interpretation $\mathcal{N}^i = (\mathcal{F}_c^i, \mathcal{F}_r^i)$ and $\mathcal{N}^j = (\mathcal{F}_c^j, \mathcal{F}_r^j)$, if there exist (unique or multiple) rigid transformations $\boldsymbol{R} \in \mathbb{SO}(3), \boldsymbol{t} \in \mathbb{R}^3$ in 3-D space such that

$$\begin{aligned} \mathcal{F}_c^i &= T\left(\mathcal{F}_r^i, \boldsymbol{R}, \boldsymbol{t}\right) \\ \mathcal{F}_c^j &= T\left(\mathcal{F}_r^j, \boldsymbol{R}, \boldsymbol{t}\right) \end{aligned} \tag{1}$$

$$\begin{aligned} T(\pi, \boldsymbol{R}, \boldsymbol{t}) &= \begin{bmatrix} \boldsymbol{R} & \boldsymbol{0}_{3\times1} \\ -\boldsymbol{t}^T\boldsymbol{R} & 1 \end{bmatrix} \begin{bmatrix} \boldsymbol{n} \\ d \end{bmatrix} \\ T(\mathcal{L}, \boldsymbol{R}, \boldsymbol{t}) &= \begin{bmatrix} \boldsymbol{R} & [\boldsymbol{t}]_\times \boldsymbol{R} \\ \boldsymbol{0}_{3\times3} & \boldsymbol{R} \end{bmatrix} \begin{bmatrix} \boldsymbol{u} \\ \boldsymbol{v} \end{bmatrix} \end{aligned} \tag{2}$$

where $T(\mathcal{F}, \boldsymbol{R}, \boldsymbol{t})$ denotes a 3-D transformation of feature $\mathcal{F}$ via $\boldsymbol{R}, \boldsymbol{t}$ defined in (2) and $[\boldsymbol{t}]_\times$ represents the skew-symmetric matrix corresponding to the vector $\boldsymbol{t}$, then $\mathcal{N}^i$ and $\mathcal{N}^j$ are internode consistent under $\boldsymbol{R}, \boldsymbol{t}$.

As seen from Definition 1, given two nodes in an interpretation, we need to solve (1). If the solution set $\mathcal{S} = \{\boldsymbol{R}, \boldsymbol{t} | \mathcal{F}_c^i = T(\mathcal{F}_r^i, \boldsymbol{R}, \boldsymbol{t}), \mathcal{F}_c^j = T(\mathcal{F}_r^j, \boldsymbol{R}, \boldsymbol{t}), \boldsymbol{R} \in \mathbb{SO}(3), \boldsymbol{t} \in \mathbb{R}^3\}$ is not empty, the nodes $\mathcal{N}^i$ and $\mathcal{N}^j$ are internode consistent according to the definition. Because the solution to $\boldsymbol{t}$ does not affect the value of $\boldsymbol{R}$, the internode consistency can be decoupled into rotation consistency and translation consistency. To this end, we first solve for the rotation consistency and then the translation consistency.

1) Rotation Consistency: For geometric features, the rigid rotation in 3-D space can be formulated as $\boldsymbol{e}_c = \boldsymbol{R}\boldsymbol{e}_r$, with $\boldsymbol{e}$ being the unit direction vector of the feature $\mathcal{F}$. Specifically, $\boldsymbol{e} = \boldsymbol{n}$ when $\mathcal{F} = \pi$ and $\boldsymbol{e} = \boldsymbol{v}$ when $\mathcal{F} = \mathcal{L}$. Known from (1) and (2), given two nodes $\mathcal{N}^i$ and $\mathcal{N}^j$, we need to solve for the rotation $\boldsymbol{R} \in \mathbb{SO}(3)$ that satisfies

$$\boldsymbol{e}_c^i = \boldsymbol{R}\boldsymbol{e}_r^i, \quad \boldsymbol{e}_c^j = \boldsymbol{R}\boldsymbol{e}_r^j. \tag{3}$$

According to the spatial configuration of the directions of features, the solution to (3) can be classified into four cases, which are presented in detail in Appendix A. Algorithm 1 gives the results of the rotation consistency and outputs the set of consistent rotations $\mathcal{R}$. Among the four cases, the 3DoF rotation can be fully constrained in three cases, except Case I, which corresponds to a special configuration of the directions of features. In the following, the translation consistency is calculated given the results of the rotation consistency.

2) Translation Consistency: Given two nodes $\mathcal{N}^i, \mathcal{N}^j$ in an interpretation and the set of consistent rotations $\mathcal{R}$, we need

Algorithm 1 Rotation Consistency

```
Input: $\{\boldsymbol{e}_c^i, \boldsymbol{e}_r^i, \boldsymbol{e}_c^j, \boldsymbol{e}_r^j\}$ – four directions of features from two nodes $\mathcal{N}^i, \mathcal{N}^j$.
Output: $\mathcal{R} = \{\boldsymbol{R} | \boldsymbol{e}_c^i = \boldsymbol{R}\boldsymbol{e}_r^i, \boldsymbol{e}_c^j = \boldsymbol{R}\boldsymbol{e}_r^j, \boldsymbol{R} \in \mathbb{SO}(3)\}$.
function RotationConsistency($\boldsymbol{e}_c^i, \boldsymbol{e}_r^i, \boldsymbol{e}_c, \boldsymbol{e}_r$)
    if $\langle \boldsymbol{e}_c^i, \boldsymbol{e}_c^j \rangle \neq \langle \boldsymbol{e}_r^i, \boldsymbol{e}_r^j \rangle$ then
        $\mathcal{R} = \emptyset$.
    else if $\boldsymbol{e}_c^i = \boldsymbol{e}_c^j$ and $\boldsymbol{e}_r^i = \boldsymbol{e}_r^j$ then
        (Case I)
        $\mathcal{R} = \{\boldsymbol{R} | \boldsymbol{R} = R(\varphi), \varphi \in \mathbb{R}\}$.
    else if $\boldsymbol{e}_c^i = \boldsymbol{e}_r^i$ and $\boldsymbol{e}_c^j = \boldsymbol{e}_r^j$ then
        (Case II)
        $\mathcal{R} = \{\boldsymbol{I}\}$.
    else if $\boldsymbol{e}_c^i = \boldsymbol{e}_r^i$ then
        (Case II)
        $\mathcal{R} = \{\mathrm{Rot}\Big(\boldsymbol{e}_c^i, \Theta(\boldsymbol{e}_c^i, \boldsymbol{e}_c^j, \boldsymbol{e}_r^j)\Big)\}$.
    else if $\boldsymbol{e}_c^j = \boldsymbol{e}_r^j$ then
        (Case II)
        $\mathcal{R} = \{\mathrm{Rot}\Big(\boldsymbol{e}_c^j, \Theta(\boldsymbol{e}_c^j, \boldsymbol{e}_c^i, \boldsymbol{e}_r^i)\Big)\}$.
    else if $(\boldsymbol{e}_c^i - \boldsymbol{e}_r^i) \times (\boldsymbol{e}_c^j - \boldsymbol{e}_r^j) = 0$ then
        (Case III)
        $\boldsymbol{r} = \boldsymbol{r}_x^i \cos\gamma_i + \boldsymbol{r}_y^i \sin\gamma_i$,
        $\theta = \frac{1}{2}\Big(\Theta(\boldsymbol{r}, \boldsymbol{e}_c^i, \boldsymbol{e}_r^i) + \Theta(\boldsymbol{r}, \boldsymbol{e}_c^j, \boldsymbol{e}_r^j)\Big)$.
        $\mathcal{R} = \{\mathrm{Rot}(\boldsymbol{r}, \theta)\}$.
    else
        (Case IV)
        $\boldsymbol{r} = \eta(\boldsymbol{e}_c^i - \boldsymbol{e}_r^i) \times (\boldsymbol{e}_c^j - \boldsymbol{e}_r^j)$,
        Compute $\theta_i = \Theta(\boldsymbol{r}, \boldsymbol{e}_c^i, \boldsymbol{e}_r^i)$ and $\theta_j = \Theta(\boldsymbol{r}, \boldsymbol{e}_c^j, \boldsymbol{e}_r^j)$.
        if $\theta_i = \theta_j$ then
            $\mathcal{R} = \{\mathrm{Rot}\big(\boldsymbol{r}, \frac{1}{2}(\theta_i + \theta_j)\big)\}$.
        else
            $\mathcal{R} = \emptyset$.
        end if
    end if
    return $\mathcal{R}$.
end function
```

to solve for the set of consistent translations $\mathcal{T}$ such that $\forall \boldsymbol{R} \in \mathcal{R}$ and $\forall \boldsymbol{t} \in \mathcal{T}$, $\mathcal{N}^i$ and $\mathcal{N}^j$ are internode consistent under $\boldsymbol{R}, \boldsymbol{t}$. Unlike the computation of the rotation consistency, different geometric features need to be handled separately when computing the translation consistency. In this article, plane and line features are adopted and three different combinations need to be considered, i.e., plane-plane case, line-line case, and plane-line case, which are detailed in Appendix B. And the results of the translation consistency are given in Algorithm 2, which outputs the final solution set $\mathcal{S} = \mathcal{R} \cup \mathcal{T}$.

Until now, both the consistent rotation and translation of the internode consistency are solved. For all $\boldsymbol{R} \in \mathcal{R}$ and $\boldsymbol{t} \in \mathcal{T}$, (1) is satisfied, i.e., the nodes $\mathcal{N}^i$ and $\mathcal{N}^j$ are internode consistent under $\boldsymbol{R}, \boldsymbol{t}$. The complete algorithmic procedure of the internode consistency is presented in Algorithm 3, which requires two nodes $\mathcal{N}^i$ and $\mathcal{N}^j$ in the IT as inputs and then outputs a set of consistent transformations $\mathcal{S}$. As can be concluded from Sections III-A1 and III-A2, according to the spatial configurations of plane and line features in 3-D space, the output $\mathcal{S}$ of Algorithm 3 takes one of the following four

Algorithm 2 Translation Consistency

```
Input: Two nodes in an interpretation $\mathcal{N}^i = (\mathcal{F}_c^i, \mathcal{F}_r^i)$, $\mathcal{N}^j = (\mathcal{F}_c^j, \mathcal{F}_r^j)$ and the set of consistent rotations $\mathcal{R}$.
Output: $\mathcal{S} = \mathcal{T} \cup \mathcal{R}' = \{\boldsymbol{t}, \boldsymbol{R}' | \mathcal{F}_c^i = T(\mathcal{F}_r^i, \boldsymbol{R}', \boldsymbol{t}), \mathcal{F}_c^j = T(\mathcal{F}_r^j, \boldsymbol{R}', \boldsymbol{t}), \boldsymbol{R}' \in \mathbb{SO}(3), \boldsymbol{t} \in \mathbb{R}^3\}$.
function TranslationConsistency($\{\mathcal{N}^i, \mathcal{N}^j\}, \mathcal{R}$)
    $\mathcal{R}' = \mathcal{R}$.
    if $\mathcal{F}_c^i = \pi_c^i, \mathcal{F}_r^i = \pi_r^i, \mathcal{F}_c^j = \pi_c^j, \mathcal{F}_r^j = \pi_r^j$ then
        (Plane-Plane case)
        if $\boldsymbol{n}_c^i = \boldsymbol{n}_c^j$ then
            if $d_r^i - d_c^i = d_r^j - d_c^j$ then
                $\mathcal{T} = \{\boldsymbol{t} | \boldsymbol{t} = \boldsymbol{t}_{pp1} + [\boldsymbol{w}_{pp1}]_\times \boldsymbol{\mu}, \boldsymbol{\mu} \in \mathbb{R}^3\}$.
            else
                $\mathcal{T} = \emptyset$.
            end if
        else if $\boldsymbol{n}_c^i \neq \boldsymbol{n}_c^j$ then
            $\mathcal{T} = \{\boldsymbol{t} | \boldsymbol{t} = \boldsymbol{t}_{pp2} + \mu \boldsymbol{w}_{pp2}, \mu \in \mathbb{R}\}$.
        end if
    else if $\mathcal{F}_c^i = \mathcal{L}_c^i, \mathcal{F}_r^i = \mathcal{L}_r^i, \mathcal{F}_c^j = \mathcal{L}_c^j, \mathcal{F}_r^j = \mathcal{L}_r^j$ then
        (Line-Line case)
        if $\boldsymbol{v}_c^i = \boldsymbol{v}_c^j$ then
            Let $\boldsymbol{e}_c = \frac{\boldsymbol{u}_c^i - \boldsymbol{u}_c^j}{\|\boldsymbol{u}_c^i - \boldsymbol{u}_c^j\|}, \boldsymbol{e}_r = \frac{\boldsymbol{u}_r^i - \boldsymbol{u}_r^j}{\|\boldsymbol{u}_r^i - \boldsymbol{u}_r^j\|}$.
            $\mathcal{R}' =$ RotationConsistency($\boldsymbol{v}_c, \boldsymbol{v}_r, \boldsymbol{e}_c, \boldsymbol{e}_r$).
            if $\|\boldsymbol{u}_r^i - \boldsymbol{u}_r^j\| = \|\boldsymbol{u}_c^i - \boldsymbol{u}_c^j\|$ && $\mathcal{R}' \neq \emptyset$ then
                $\mathcal{T} = \{\boldsymbol{t} | \boldsymbol{t} = \boldsymbol{t}_{ll1} + \mu \boldsymbol{w}_{ll1}, \mu \in \mathbb{R}\}$.
            else
                $\mathcal{T} = \emptyset$.
            end if
        else if $\boldsymbol{v}_c^i \neq \boldsymbol{v}_c^j$ then
            if $l(\mathcal{L}_r^i, \mathcal{L}_r^j) - l(\mathcal{L}_c^i, \mathcal{L}_c^j) = 0$ then
                $\mathcal{T} = \{(\boldsymbol{A}_{ll}^T \boldsymbol{A}_{ll})^{-1} \boldsymbol{A}_{ll}^T \boldsymbol{b}_{ll}\}$.
            else
                $\mathcal{T} = \emptyset$.
            end if
        end if
    else if $\mathcal{F}_c^i = \pi_c^i, \mathcal{F}_r^i = \pi_r^i, \mathcal{F}_c^j = \mathcal{L}_c^j, \mathcal{F}_r^j = \mathcal{L}_r^j$ then
        (Plane-Line case)
        if ${\boldsymbol{n}_c^i}^T \boldsymbol{v}_c^j = 0$ then
            if $l(\pi_r^i, \mathcal{L}_r^j) - l(\pi_c^i, \mathcal{L}_c^j) = 0$ then
                $\mathcal{T} = \{\boldsymbol{t} | \boldsymbol{t} = \boldsymbol{t}_{pl1} + \mu \boldsymbol{w}_{pl1}, \mu \in \mathbb{R}\}$.
            else
                $\mathcal{T} = \emptyset$.
            end if
        else if $\boldsymbol{n}_c^i = \boldsymbol{v}_c^j$ then
            $\mathcal{T} = \{\boldsymbol{t} | \boldsymbol{t} = \boldsymbol{t}_{pl2} + \boldsymbol{R}\boldsymbol{w}_{pl2}, \boldsymbol{R} \in \mathcal{R}\}$.
        else
            $\mathcal{T} = \{(\boldsymbol{A}_{pl}^T \boldsymbol{A}_{pl})^{-1} \boldsymbol{A}_{pl}^T \boldsymbol{b}_{pl}\}$.
        end if
    end if
    return $\mathcal{S} = \mathcal{T} \cup \mathcal{R}'$.
end function
```

TABLE II

SPATIAL CONFIGURATIONS OF FEATURES CORRESPONDING TO DIFFERENT FORMS OF THE TRANSFORMATION SET

	plane-plane	line-line	plane-line
1rDoF&2tDoFs (unconstrained)	$\boldsymbol{n}_c^i = \boldsymbol{n}_c^j$	–	–
1rDoF(unconstrained)	–	–	$\boldsymbol{n}_c^i = \boldsymbol{v}_c^j$
1tDoF(unconstrained)	$\boldsymbol{n}_c^i \neq \boldsymbol{n}_c^j$	$\boldsymbol{v}_c^i = \boldsymbol{v}_c^j$	${\boldsymbol{n}_c^i}^T \boldsymbol{v}_c^j = 0$
Constrained	–	$\boldsymbol{v}_c^i \neq \boldsymbol{v}_c^j$	$\boldsymbol{n}_c^i \neq \boldsymbol{v}_c^j$ ${\boldsymbol{n}_c^i}^T \boldsymbol{v}_c^j \neq 0$

forms (rDoF and tDoF are abbreviations for rotational and translational DoFs, respectively).

1) *1 rDoF and* 2 *tDoFs:* $\mathcal{S} = \{\boldsymbol{R}, \boldsymbol{t} | \boldsymbol{R} = R(\varphi), \varphi \in \mathbb{R}, \boldsymbol{t} = \boldsymbol{t}_0 + [\boldsymbol{w}]_\times \boldsymbol{\mu}, \boldsymbol{\mu} \in \mathbb{R}^3\}$.
2) *1 rDoF:* $\mathcal{S} = \{\boldsymbol{R}, \boldsymbol{t} | \boldsymbol{R} = R(\varphi), \boldsymbol{t} = \boldsymbol{t}_0 + \boldsymbol{R}\boldsymbol{w}, \varphi \in \mathbb{R}\}$.
3) *1 tDoF:* $\mathcal{S} = \{\boldsymbol{R}, \boldsymbol{t} | \boldsymbol{R} = \mathrm{Rot}(\boldsymbol{r}, \theta), \boldsymbol{t} = \boldsymbol{t}_0 + \mu \boldsymbol{w}, \mu \in \mathbb{R}\}$.
4) *Constrained:* $\mathcal{S} = \{\boldsymbol{R}, \boldsymbol{t} | \boldsymbol{R} = \mathrm{Rot}(\boldsymbol{r}, \theta), \boldsymbol{t} = \boldsymbol{t}_0\}$.

The spatial configurations of geometric features corresponding to the four forms are listed in Table II. For the general configurations of both the line-line case ($\boldsymbol{v}_c^i \neq \boldsymbol{v}_c^j$) and plane-line case ($\boldsymbol{n}_c^i \neq \boldsymbol{v}_c^j, {\boldsymbol{n}_c^i}^T \boldsymbol{v}_c^j \neq 0$), the transformation can be fully constrained by two nodes. As for the unconstrained cases, there are infinite solutions to the transformation and the explicit expression is given in Algorithm 3.

Algorithm 3 Internode Consistency

```
Input: Two nodes $\mathcal{N}^i = (\mathcal{F}_c^i, \mathcal{F}_r^i), \mathcal{N}^j = (\mathcal{F}_c^j, \mathcal{F}_r^j)$.
Output: $\mathcal{S} = \{\boldsymbol{R}, \boldsymbol{t} | \mathcal{F}_c^i = T(\mathcal{F}_r^i, \boldsymbol{R}, \boldsymbol{t}), \mathcal{F}_c^j = T(\mathcal{F}_r^j, \boldsymbol{R}, \boldsymbol{t})\}$
function InterNodeConsistency($\{\mathcal{N}^i, \mathcal{N}^j\}$)
    if $\mathcal{F}_c^i = \pi_c^i, \mathcal{F}_r^i = \pi_r^i, \mathcal{F}_c^j = \pi_c^j, \mathcal{F}_r^j = \pi_r^j$ then
        $\boldsymbol{e}_c^i = \boldsymbol{n}_c^i, \boldsymbol{e}_r^i = \boldsymbol{n}_r^i, \boldsymbol{e}_c^j = \boldsymbol{n}_c^j, \boldsymbol{e}_r^j = \boldsymbol{n}_r^j$.
    else if $\mathcal{F}_c^i = \mathcal{L}_c^i, \mathcal{F}_r^i = \mathcal{L}_r^i, \mathcal{F}_c^j = \mathcal{L}_c^j, \mathcal{F}_r^j = \mathcal{L}_r^j$ then
        $\boldsymbol{e}_c^i = \boldsymbol{v}_c^i, \boldsymbol{e}_r^i = \boldsymbol{v}_r^i, \boldsymbol{e}_c^j = \boldsymbol{v}_c^j, \boldsymbol{e}_r^j = \boldsymbol{v}_r^j$.
    else if $\mathcal{F}_c^i = \pi_c^i, \mathcal{F}_r^i = \pi_r^i, \mathcal{F}_c^j = \mathcal{L}_c^j, \mathcal{F}_r^j = \mathcal{L}_r^j$ then
        $\boldsymbol{e}_c^i = \boldsymbol{n}_c^i, \boldsymbol{e}_r^i = \boldsymbol{n}_r^i, \boldsymbol{e}_c^j = \boldsymbol{v}_c^j, \boldsymbol{e}_r^j = \boldsymbol{v}_r^j$.
    end if
    $\mathcal{R}$ =RotationConsistency($\boldsymbol{e}_c^i, \boldsymbol{e}_r^i, \boldsymbol{e}_c^j, \boldsymbol{e}_r^j$).
    $\mathcal{S}$ =TranslationConsistency($\{\mathcal{N}^i, \mathcal{N}^j\}, \mathcal{R}$).
    return $\mathcal{S}$.
end function
```

In the process of tree expansion, for a new node $\mathcal{N}^{n+1}$ generated by a hypothesized association of geometric features and an existing interpretation $\mathcal{P}_n$ in the IT, we compute the internode consistency between $\mathcal{N}^{n+1}$ and each node in $\mathcal{P}_n$ by $\mathcal{S}_{(n+1)i} = \textsc{InterNodeConsistency}(\{\mathcal{N}^{n+1}, \mathcal{N}^i\}), \forall i \in \{1, \ldots, n\}$. If $\mathcal{S}_{(n+1)i} \neq \emptyset, \forall i \in \{1, \ldots, n\}$, the new node $\mathcal{N}^{n+1}$ is added to $\mathcal{P}_n$, which results in an $(n+1)$-interpretation $\mathcal{P}_{n+1} = \{\mathcal{N}^{n+1}\} \cup \mathcal{P}_n$. After $\mathcal{P}_{n+1}$ is generated, for any two nodes $\mathcal{N}^i$ and $\mathcal{N}^j$ $(\forall i, j \in \{1, \ldots, n+1\})$ in $\mathcal{P}_{n+1}$, the two feature pairs corresponding to $\mathcal{N}^i$ and $\mathcal{N}^j$, respectively, can be aligned by any $\boldsymbol{R}, \boldsymbol{t} \in \mathcal{S}_{ij} = \textsc{InterNodeConsistency}(\{\mathcal{N}^i, \mathcal{N}^j\})$. However, $\forall i, j, k, l \in \{1, \ldots, n+1\} (i \neq j \neq k \neq l)$, $\mathcal{S}_{ij}$ and $\mathcal{S}_{kl}$ are possibly not equivalent to each other (see Fig. 3 as an example). Therefore, computation of the internode consistency cannot give the final transformations that align the associated features corresponding to *all the nodes* in the interpretation. Therefore, when an interpretation is updated by adding a new node, we need to fuse the outputs of the internode consistency of any two nodes in the interpretation and solve for the feasible transformations corresponding to the whole interpretation. To this end, the CTM for an interpretation is defined and updated in Section III-B.

Algorithm 4 CTM Update

```
Input: The CTM $\mathcal{M}(\mathcal{P}_n, \mathcal{S}_n)$ for the $n$-interpretation $\mathcal{P}_n$ and
    a newly added node $\mathcal{N}^{n+1}$.
Output: The updated CTM $\mathcal{M}(\mathcal{P}_{n+1}, \mathcal{S}_{n+1})$ for the $(n+1)$-
    interpretation $\mathcal{P}_{n+1}$.
1:  function Update($\mathcal{M}(\mathcal{P}_n, \mathcal{S}_n), \mathcal{N}^{n+1}$)
2:      $\mathcal{P}_{n+1} = \{\mathcal{N}^{n+1}\} \cup \mathcal{P}_n$.
3:      $\mathcal{S} = \mathcal{S}_n$.
4:      for $i = 1$ to $n$ do
5:          $\mathcal{S}_{(n+1)i}$ =InterNodeConsistency($\{\mathcal{N}^{n+1}, \mathcal{N}^i\}$).
6:          if $\mathcal{S}_{(n+1)i} = \emptyset$ then
7:              $\mathcal{S} = \emptyset$ and break.
8:          else
9:              $\mathcal{S}$ =Fuse($\mathcal{S}, \mathcal{S}_{(n+1)i}$).
10:         end if
11:     end for
12:     $\mathcal{S}_{n+1} = \mathcal{S}$.
13:     return $\mathcal{M}(\mathcal{P}_{n+1}, \mathcal{S}_{n+1})$.
14: end function
```

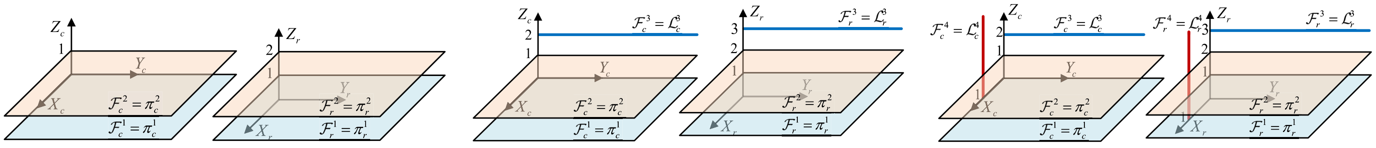
B. CTM

The CTM is defined for each interpretation in the IT, which consists of the consistent transformations that align the associated features represented by all the nodes in the interpretation. The CTM is updated when a new node is added to an interpretation, fusing the outputs of the internode consistency. During expansion of the IT, as more nodes are added to an interpretation, more constraints are available for estimation of the camera pose, that is to say, the set of consistent transformations is getting smaller. Fig. 3 gives a very simple example to illustrate this process.

Definition 2 (CTM): For an n-interpretation $\mathcal{P}_n = \{\mathcal{N}^n, \mathcal{N}^{n-1}, \ldots, \mathcal{N}^1\}$, a CTM is defined by $\mathcal{M}(\mathcal{P}_n, \mathcal{S}_n)$, with $\mathcal{S}_n = \{\boldsymbol{R}, \boldsymbol{t} | \mathcal{F}_c^i = T(\mathcal{F}_r^i, \boldsymbol{R}, \boldsymbol{t}), \forall i = 1, \ldots, n\}$. If $\mathcal{S}_n \neq \emptyset$, $\forall \boldsymbol{R}, \boldsymbol{t} \in \mathcal{S}_n$, the n-interpretation $\mathcal{P}_n$ is **consistent** under $\boldsymbol{R}, \boldsymbol{t}$.

The CTM for each interpretation is maintained during IT expansion and the specific procedure of CTM update is given in Algorithm 4. When a new node $\mathcal{N}^{n+1}$ is added to $\mathcal{P}_n$ yielding $\mathcal{P}_{n+1}$, the updated CTM $\mathcal{M}(\mathcal{P}_{n+1}, \mathcal{S}_{n+1})$ can be computed by $\mathcal{M}(\mathcal{P}_{n+1}, \mathcal{S}_{n+1}) = \textsc{Update}\big(\mathcal{M}(\mathcal{P}_n, \mathcal{S}_n), \mathcal{N}^{n+1}\big)$.

Specifically, $\mathcal{P}_{n+1}$ is simply the union of $\mathcal{P}_n$ and $\{\mathcal{N}^{n+1}\}$, as in line 2 of Algorithm 4. And $\mathcal{S}_{n+1}$ is updated by the function Fuse, which solves the intersection of $\mathcal{S}_n$ and $\mathcal{S}_{(n+1)i}, \forall i \in \{1, \ldots, n\}$, as in lines 4–11. The detailed


Z_c
Y_c
X_c
$\mathcal{F}_c^2 = \pi_c^2$
$\mathcal{F}_c^1 = \pi_c^1$
Z_r
Y_r
X_r
$\mathcal{F}_r^2 = \pi_r^2$
$\mathcal{F}_r^1 = \pi_r^1$
$\mathcal{F}_c^3 = \mathcal{L}_c^3$
$\mathcal{F}_r^3 = \mathcal{L}_r^3$
$\mathcal{F}_c^4 = \mathcal{L}_c^4$
$\mathcal{F}_r^4 = \mathcal{L}_r^4$
1
2
3

(a) (b) (c)

Fig. 3. (Simple example.) (a) For nodes $\mathcal{N}^1 = (\mathcal{F}_c^1, \mathcal{F}_r^1)$ and $\mathcal{N}^2 = (\mathcal{F}_c^2, \mathcal{F}_r^2)$, the result of internode consistency is $\mathcal{S}_{21} = \{\boldsymbol{R}, \boldsymbol{t} | \boldsymbol{R} = \mathrm{Rot}(\boldsymbol{r}, \varphi), \varphi \in \mathbb{R}, \boldsymbol{t} = \boldsymbol{t}_0 + [\boldsymbol{r}]_\times \boldsymbol{\mu}, \boldsymbol{\mu} \in \mathbb{R}^3\}$, with $\boldsymbol{r} = [0, 0, 1]^T$ and $\boldsymbol{t}_0 = [0, 0, -1]^T$. Note that the 1DoF rotation around $\boldsymbol{r}$ and the 2DoF translation on the plane vertical to $\boldsymbol{r}$ cannot be constrained by the two pairs of features. The CTM for the 2-interpretation $\mathcal{P}_2 = \{\mathcal{N}^2, \mathcal{N}^1\}$ is $\mathcal{M}(\mathcal{P}_2, \mathcal{S}_2)$, with $\mathcal{S}_2 = \mathcal{S}_{21}$. (b) For nodes $\mathcal{N}^3 = (\mathcal{F}_c^3, \mathcal{F}_r^3)$ and $\mathcal{N}^2$, the result of internode consistency is $\mathcal{S}_{32} = \{\boldsymbol{R}, \boldsymbol{t} | \boldsymbol{R} = \boldsymbol{I}, \boldsymbol{t} = \boldsymbol{t}_0 + \mu \boldsymbol{w}_0, \mu \in \mathbb{R}\}$, with $\boldsymbol{w} = [0, 1, 0]^T$. For nodes $\mathcal{N}^3$ and $\mathcal{N}^1$, $\mathcal{S}_{31} = \mathcal{S}_{32}$. The CTM for the 3-interpretation $\mathcal{P}_2 = \{\mathcal{N}^3, \mathcal{P}^2\}$ is $\mathcal{M}(\mathcal{P}_3, \mathcal{S}_3)$, with $\mathcal{S}_3 = \{\boldsymbol{R}, \boldsymbol{t} | \boldsymbol{R} = \boldsymbol{I}, \boldsymbol{t} = \boldsymbol{t}_0 + \mu \boldsymbol{w}_0, \mu \in \mathbb{R}\}$ (determination of $\mathcal{S}_3$ is detailed in Algorithm 4). Note that the unconstrained 1DoF rotation and one of the two DoFs of the translation in (a) are constrained by adding a new node $\mathcal{N}^3$. The 1DoF translation along $\boldsymbol{w}$ still cannot be constrained. (c) For nodes $\mathcal{N}^4 = (\mathcal{F}_c^4, \mathcal{F}_r^4)$ and $\mathcal{N}^3$, the result of internode consistency is $\mathcal{S}_{43} = \{\boldsymbol{R}, \boldsymbol{t} | \boldsymbol{R} = \boldsymbol{I}, \boldsymbol{t} = \boldsymbol{t}_0\}$. For nodes $\mathcal{N}^4$ and $\mathcal{N}^2$, $\mathcal{S}_{42} = \{\boldsymbol{R}, \boldsymbol{t} | \boldsymbol{R} = \mathrm{Rot}(\boldsymbol{r}, \varphi), \boldsymbol{t} = \boldsymbol{t}_1 + \boldsymbol{R}\boldsymbol{w}_1, \varphi \in \mathbb{R}\}$, with $\boldsymbol{t}_1 = [1, 0, -1]^T$ and $\boldsymbol{w}_1 = [-1, 0, 0]^T$. For nodes $\mathcal{N}^4$ and $\mathcal{N}^1$, $\mathcal{S}_{41}$ is the same as $\mathcal{S}_{42}$. The CTM for the 4-interpretation $\mathcal{P}_4 = \{\mathcal{N}^4, \mathcal{P}^3\}$ is $\mathcal{M}(\mathcal{P}_4, \mathcal{S}_4)$, with $\mathcal{S}_4 = \{\boldsymbol{R}, \boldsymbol{t} | \boldsymbol{R} = \boldsymbol{I}, \boldsymbol{t} = \boldsymbol{t}_0\}$. It is obvious that until here the transformation between the two frames is completely constrained by the four associated features.

procedure of the function FUSE in line 9 is given in the Supplementary Material. The consistent transformation set for an interpretation takes one of the four forms as is listed in Table II. For each form, the function FUSE directly outputs the intersection of two transformation sets, as detailed in the Supplementary Material.

After the CTM update in Algorithm 4, the newly added node $\mathcal{N}^{n+1}$ is consistent with all the nodes in $\mathcal{P}_n$ under $\boldsymbol{R}, \boldsymbol{t} \in \mathcal{S}_{n+1}$. Only if the resultant $\mathcal{S}_{n+1} \neq \emptyset$, $\mathcal{P}_{n+1}$ is accepted as an interpretation in the IT and is continually updated in the subsequent expansion of the tree. Otherwise, if $\mathcal{S}_{n+1} = \emptyset$, $\mathcal{P}_{n+1}$ is pruned from the tree. When the nodes in an interpretation are insufficient to constrain the transformation, there are infinite solutions to the CTM. With the incremental update of the CTM, more constraints are added to the unconstrained estimate until a unique solution is obtained (see Fig. 3 as an intuitive example). It is worth pointing out that in most circumstances, very few nodes are sufficient to fully constrain the transformation estimate (e.g., for some configurations, only two nodes are sufficient, as in Table II) and the constrained estimate can be further confirmed by more nodes during update of the CTM. Thus, the robustness of the algorithm against outliers and noises can be largely increased. Compared with the traditional pruning strategy for the IT structure [25]–[27], the proposed method solves the problems of feature association and robot localization simultaneously through the calculation of internode consistency. The constrained subspace of robot poses is computed and explicitly represented by the incremental update of CTM. Furthermore, the spatial configuration of geometric features can also be represented, which is beneficial to the active navigation in the environment. After the IT is constructed, the interpretation with the greatest number of nodes is chosen as the final result of the feature association. In the implementation, a null node is added to each node as a wild card, in case that there exist features in the current frame that do not have any correspondences in the reference frame.

To the best of our knowledge, this is the first feature association method that combines two different geometric features into one unified framework. Theoretically, the framework is readily extensible to any combination of different types of geometric features that are properly parameterized. Compared with the systems using only one type of feature, the hybrid feature framework exploits the advantages of different features. For instance, plane features are more stable than line features. However, when estimating camera poses using plane features, degeneracy often occurs due to insufficient quantity of planes extracted from an RGB-D image. In comparison, the quantity of the extracted line features is usually much greater, and thus can provide sufficient constraints for the pose estimation in most situations. Theoretically, two nonparallel lines are enough to constrain the pose estimation, as can be seen in Table II. In our method, the complementary advantages of planes and lines are fully exploited and improvement of the performance is proven by the experiments. Furthermore, unlike other multifeature methods using NN search-based association [7], [11], [13] or RANSAC-based association [12], [14], the associated features in our method are ensured to be consistent under a common transformation model which is incrementally updated in a closed form. Thus, the resultant feature pairs and camera pose estimate can be applied in the subsequent optimization process without any inconsistency, and the realtime performance can also be guaranteed.

IV. HYBRID FEATURE JOINT OPTIMIZATION

In Section III, we have performed the geometric feature matching between the current and reference frames. The associated planes and lines are denoted by $\{\pi_{ci}, \pi_{ri}\}_{i=1,\ldots,N_\pi}$ and $\{\mathcal{L}_{cj}, \mathcal{L}_{rj}\}_{j=1,\ldots,N_\mathcal{L}}$, respectively. In the meantime, an estimate of the transformation $\boldsymbol{R}, \boldsymbol{t}$ of the RGB-D camera is obtained. Nevertheless, the planes and lines are fitted directly by the noisy measurements. Uncertainties from feature extraction are involved in the camera pose estimation. In this section, the transformation of the camera as well as the geometric landmarks in the map are further refined by a hybrid feature joint optimization process.

The overall cost function of the optimization is

$$F(\boldsymbol{R}, \boldsymbol{t}, \tilde{\pi}_{ri}, \tilde{\mathcal{L}}_{rj}) = \sum_{i=1}^{N_\pi} \left\| h_\pi\left(\boldsymbol{R}, \boldsymbol{t}, \tilde{\pi}_{ri}\right) - \pi_{ci} \right\|^2_{\boldsymbol{C}_{\pi_{ci}}} + \sum_{j=1}^{N_\mathcal{L}} \left\| h_\mathcal{L}\left(\boldsymbol{R}, \boldsymbol{t}, \tilde{\mathcal{L}}_{rj}\right) - \mathcal{L}_{cj} \right\|^2_{\boldsymbol{C}_{\mathcal{L}_{cj}}} \quad (4)$$

where $\tilde{\pi}_r$ and $\tilde{\mathcal{L}}_r$ represent the plane and line landmarks in the map, respectively, described in the reference frame. And $h_\pi(\boldsymbol{R}, \boldsymbol{t}, \tilde{\pi}_{ri})$ and $h_{\mathcal{L}}(\boldsymbol{R}, \boldsymbol{t}, \tilde{\mathcal{L}}_{rj})$ are the measurement models corresponding to the planes and lines, respectively, which are given in Sections IV-A and IV-B. Note that both the plane and line features are overparameterized in Section III. In 3-D space, a plane has three DoFs and a line has four DoFs, but they are parameterized by a 4-vector and 6-vector, respectively. To this end, in Sections IV-A and IV-B, minimal representations are used to update the plane and line parameters in optimization, respectively. In addition, the covariance matrices $\boldsymbol{C}_{\pi_{ci}}$ and $\boldsymbol{C}_{\mathcal{L}_{cj}}$ of planes and lines, respectively, are computed in Section IV-C. They are used not only to decrease the effect of uncertainties, but also to balance the contributions of two types of features represented in different parameter spaces.

A. Parameterization of Planes

In Section III, a plane extracted from the RGB-D images is parameterized by $\pi = [\boldsymbol{n}^T, d]^T$ with $\boldsymbol{n}^T\boldsymbol{n} = 1$. If a point $\boldsymbol{p} \in \mathbb{R}^3$ in 3-D space is on the plane π, it satisfies $\boldsymbol{n}^T\boldsymbol{p}+d = 0$. Note that the equation cannot be affected by multiplication by a non-zero scalar. Thus, a plane in 3-D space has only three DoFs. The homogeneous representation of the plane is denoted by $\Pi \in \mathbb{P}^3$ in projective space [23]. Spherically normalizing the homogeneous vector Π yields $\tilde{\pi} = \Pi/\|\Pi\| \in \mathbb{S}^3$, where $\mathbb{S}^3$ is the 3-sphere in the space $\mathbb{R}^4$, which is a Lie group under the operation of quaternion multiplication. During optimization, the exponential map from $\mathbb{R}^3$ to $\mathbb{S}^3$ [46] is used to update a plane $\tilde{\pi} \in \mathbb{S}^3$ by an increment $\zeta \in \mathbb{R}^3$ using the quaternion multiplication $\tilde{\pi}' = \exp(\zeta) \circ \tilde{\pi}$. The measurement model is

$$h_\pi\big(\boldsymbol{R}, \boldsymbol{t}, \tilde{\pi}_r\big) = \begin{bmatrix} \boldsymbol{R} & \boldsymbol{0}_{3\times 1} \\ -\boldsymbol{t}^T\boldsymbol{R} & 1 \end{bmatrix} \cdot \frac{\tilde{\pi}_r}{\|\tilde{\boldsymbol{n}}_r\|} \tag{5}$$

where $\tilde{\pi}_r = [\tilde{\boldsymbol{n}}_r^T, \tilde{d}_r]^T$ is the plane to be updated, which is described in the reference coordinate system, and $(\boldsymbol{R}, \boldsymbol{t}) \in \mathbb{SE}(3)$ is the rigid body transformation from the reference coordinate system to the current one.

B. Parameterization of Lines

For the parameterization of lines, we use two forms of line representations as in [41], i.e., the Plücker coordinates for a global parameterization of line features and the orthonormal representation for the local update during optimization. The Plücker coordinates are denoted by $\tilde{\mathcal{L}} = [\tilde{\boldsymbol{u}}^T, \tilde{\boldsymbol{v}}^T]^T \in \mathbb{P}^5$ satisfying the Plücker constraint $\tilde{\boldsymbol{u}}^T\tilde{\boldsymbol{v}} = 0$. $\tilde{\boldsymbol{u}} \in \mathbb{R}^3$ is normal to the plane defined by the join of the line and the origin, and $\tilde{\boldsymbol{v}} \in \mathbb{R}^3$ is the line direction. And the orthonormal representation [42] is denoted by $(\boldsymbol{Q}, \boldsymbol{W}) \in \mathbb{SO}(3) \times \mathbb{SO}(2)$. The orthonormal representation $(\boldsymbol{Q}, \boldsymbol{W})$ of a 3-D line $\tilde{\mathcal{L}}$ can be computed using a QR decomposition $[\tilde{\boldsymbol{u}}|\tilde{\boldsymbol{v}}] = \boldsymbol{Q}\boldsymbol{\Sigma}$, $\boldsymbol{\Sigma} = \text{diag}(\sigma_1, \sigma_2)$. The matrix $\boldsymbol{W}$ is set to

$$\boldsymbol{W} = \begin{bmatrix} \sigma_1 & -\sigma_2 \\ \sigma_2 & \sigma_1 \end{bmatrix} \in \mathbb{SO}(2). \tag{6}$$

The minimum four parameters for a line are denoted by $\boldsymbol{\rho} = [\boldsymbol{\phi}^T, \phi]^T$, $\boldsymbol{\phi} \in \mathbb{R}^3$, $\phi \in \mathbb{R}$. $(\boldsymbol{Q}, \boldsymbol{W})$ is updated by $\boldsymbol{Q} = \boldsymbol{Q}\exp([\boldsymbol{\phi}]_\times)$ and $\boldsymbol{W} = \boldsymbol{W}\exp([\phi]_\times)$, where $[\boldsymbol{\phi}]_\times$ and $[\phi]_\times$ are the 3×3 and 2×2 skew symmetric matrices corresponding to $\boldsymbol{\phi}$ and ϕ, respectively. And $(\boldsymbol{Q}, \boldsymbol{W})$ can be converted to the Plücker coordinates $\tilde{\mathcal{L}}$ by $\tilde{\mathcal{L}} = [\sigma_1\boldsymbol{q}_1^T, \sigma_2\boldsymbol{q}_2^T]^T$, where $\boldsymbol{q}_i$ is the ith column of $\boldsymbol{Q}$.

The measurement model of the line is represented by

$$h_{\mathcal{L}}\big(\boldsymbol{R}, \boldsymbol{t}, \tilde{\mathcal{L}}_r\big) = \begin{bmatrix} \boldsymbol{R} & [\boldsymbol{t}]_\times\boldsymbol{R} \\ \boldsymbol{0} & \boldsymbol{R} \end{bmatrix} \cdot \frac{\tilde{\mathcal{L}}_r}{\tilde{\boldsymbol{v}}_r} \tag{7}$$

where $\tilde{\mathcal{L}}_r = [\tilde{\boldsymbol{u}}_r^T, \tilde{\boldsymbol{v}}_r^T]^T$ is the line to be updated, which is described in the reference coordinate system.

C. Computation of Covariances

The extracted plane $\pi = [\boldsymbol{n}^T, d]^T$ is computed by minimizing the sum of squared distances from observed data points $\boldsymbol{p}_\pi$ on the plane to the fitted plane model

$$E_\pi(\boldsymbol{n}, d) = \frac{1}{2}\sum_{i=1}^{N_{p\pi}}\big(\boldsymbol{n}^T\boldsymbol{p}_{\pi i} + d\big)^2, \quad \text{s. t. } \boldsymbol{n}^T\boldsymbol{n} = 1. \tag{8}$$

$N_{p\pi}$ represents the number of points that are supposed to fit a plane model. By taking the partial derivative of $E_\pi(\boldsymbol{n}, d)$ with respect to d and setting it to zero, the optimal estimate of d can be computed by

$$d^* = -\boldsymbol{n}^T\boldsymbol{p}_{G\pi} \tag{9}$$

with $\boldsymbol{p}_{G\pi} = \frac{1}{N_{p\pi}}\sum_{i=1}^{N_{p\pi}}\boldsymbol{p}_{\pi i}$. Substituting (9) into (8) yields

$$E_\pi(\boldsymbol{n}) = \frac{1}{2}\boldsymbol{n}^T\boldsymbol{S}_\pi\boldsymbol{n} \tag{10}$$

where $\boldsymbol{S}_\pi = \sum_{i=1}^{N_{p\pi}}(\boldsymbol{p}_{\pi i} - \boldsymbol{p}_{G\pi})(\boldsymbol{p}_{\pi i} - \boldsymbol{p}_{G\pi})^T$. The optimal estimate $\boldsymbol{n}^*$ equals the eigenvector of $\boldsymbol{S}_\pi$ corresponding to the smallest eigenvalue. The pseudoinverse of the covariance of $(\boldsymbol{n}, d)$ is estimated by the Hessian matrix

$$\boldsymbol{C}_\pi^\dagger = \boldsymbol{H}_\pi|_{\boldsymbol{n}^*, d^*} = \sum_{i=1}^{N_{p\pi}}\begin{bmatrix} \boldsymbol{p}_{\pi i}\boldsymbol{p}_{\pi i}^T & \boldsymbol{p}_{\pi i} \\ \boldsymbol{p}_{\pi i}^T & 1 \end{bmatrix}. \tag{11}$$

Likewise, the extracted line $\mathcal{L} = [\boldsymbol{u}^T, \boldsymbol{v}^T]^T$ is computed by minimizing the sum of squared distances from observed data points $\boldsymbol{p}_{\mathcal{L}}$ on the line to the fitted line model

$$E_{\mathcal{L}}(\boldsymbol{u}, \boldsymbol{v}) = \frac{1}{2}\sum_{i=1}^{N_{p\mathcal{L}}}\big\|\boldsymbol{u} - [\boldsymbol{p}_{\mathcal{L}i}]_\times\boldsymbol{v}\big\|^2$$

$$\text{s. t. } \boldsymbol{v}^T\boldsymbol{v} = 1, \quad \boldsymbol{u}^T\boldsymbol{v} = 0. \tag{12}$$

$N_{p\mathcal{L}}$ represents the number of points that are supposed to fit a line model. By taking the partial derivative of $E_{\mathcal{L}}(\boldsymbol{u}, \boldsymbol{v})$ with respect to $\boldsymbol{u}$ and setting it to zero, the optimal estimate of $\boldsymbol{u}$ can be computed by

$$\boldsymbol{u}^* = [\boldsymbol{p}_{G\mathcal{L}}]_\times\boldsymbol{v} \tag{13}$$

with $[\boldsymbol{p}_{G\mathcal{L}}]_\times = \frac{1}{N_{p\mathcal{L}}}\sum_{i=1}^{N_{p\mathcal{L}}}[\boldsymbol{p}_{\mathcal{L}i}]_\times$. Substituting (13) into (12) yields

$$E_{\mathcal{L}}(\boldsymbol{v}) = \frac{1}{2}\boldsymbol{v}^T\boldsymbol{S}_{\mathcal{L}}\boldsymbol{v} \tag{14}$$

where $\boldsymbol{S}_{\mathcal{L}} = \sum_{i=1}^{N_{p\mathcal{L}}} [\boldsymbol{p}_{\mathcal{L}i} - \boldsymbol{p}_{G\mathcal{L}}]_\times^T [\boldsymbol{p}_{\mathcal{L}i} - \boldsymbol{p}_{G\mathcal{L}}]_\times$. The optimal estimate $\boldsymbol{v}^*$ also equals the eigenvector of $\boldsymbol{S}_{\mathcal{L}}$ corresponding to the smallest eigenvalue, and the pseudoinverse of the covariance of $(\boldsymbol{u}, \boldsymbol{v})$ is

$$\boldsymbol{C}_{\mathcal{L}}^{\dagger} = \boldsymbol{H}_{\mathcal{L}}|_{\boldsymbol{u}^*,\boldsymbol{v}^*} = \sum_{i=1}^{N_{p\mathcal{L}}} \begin{bmatrix} \boldsymbol{I}_3 & [\boldsymbol{p}_{\mathcal{L}i}]_\times^T \\ [\boldsymbol{p}_{\mathcal{L}i}]_\times & [\boldsymbol{p}_{\mathcal{L}i}]_\times^T [\boldsymbol{p}_{\mathcal{L}i}]_\times \end{bmatrix}. \quad (15)$$

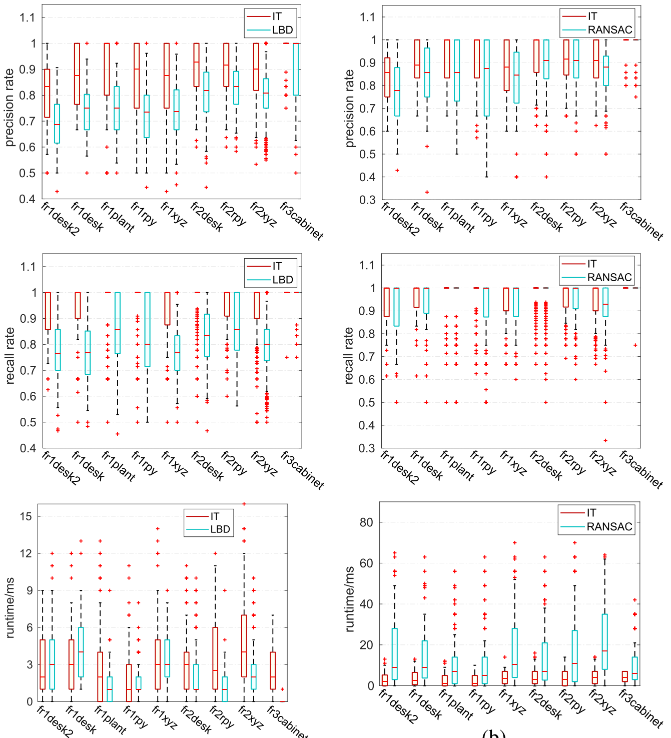
V. Experiments

In this section, extensive experiments are performed over the TUM [47] and ICL-NUIM [48] benchmarks. Specifically, the precise and recall rates of the proposed hybrid feature association method are evaluated in Section V-A and the translational and rotational errors of the frame-to-frame registration are compared in Section V-B. In Section V-D, the proposed IT-HYFAO-VO is compared with other three VO methods. In Section V-E, a complete SLAM system composed of IT-HYFAO-VO and a general factor-graph-based back-end optimization is compared with eight state-of-the-art SLAM systems. The test platform for all the experiments is a computer with an Intel i7 CPU at 1.8 GHz and 8G RAM.

A. Experiments on Feature Association

The proposed IT-based hybrid feature association method is compared with the matching methods using the visual descriptor and RANSAC scheme, respectively. We manually label the corresponding feature pairs in successive frames, which are used as the groundtruth in the experiment. The precision rate, recall rate, and the runtime of each method are evaluated and compared. Precision rate is defined as the number of associated features that are labeled divided by the total number of associated features. Recall rate is the number of associated features that are labeled divided by the total number of labeled features.

We first compare the IT-based method with the visual descriptor-based matching method, in which the LBD [21] is used. The LBD is an effective line descriptor that is widely employed in VO and SLAM systems [7], [11], [13]. For the sake of fairness, the IT-based association is run with only the line features in this experiment. The two methods are performed over nine image sequences from the TUM dataset. It can be seen from Fig. 4(a) that both the precision and recall rates of the IT-based line matching are higher than those of the LBD-based line matching. The reason is that the LBD is based on the visual appearance around the 2-D line on the image. It is largely affected by lighting conditions as well as the motion blurs, which is a common disadvantage of most visual features. Moreover, the LBD-based line matching does not take into account the geometric information of environments, which presents much higher robustness and stability than point features. In contrast, the proposed IT-based line matching fully exploits the high-level geometric features in 3-D spaces, which are independent of the visual appearance, and thus insensitive to the location and orientation of the camera. The bottom figure of Fig. 4(a) shows the runtime of the frame-to-frame association of the two methods. It can be seen that both methods are quite time-saving.



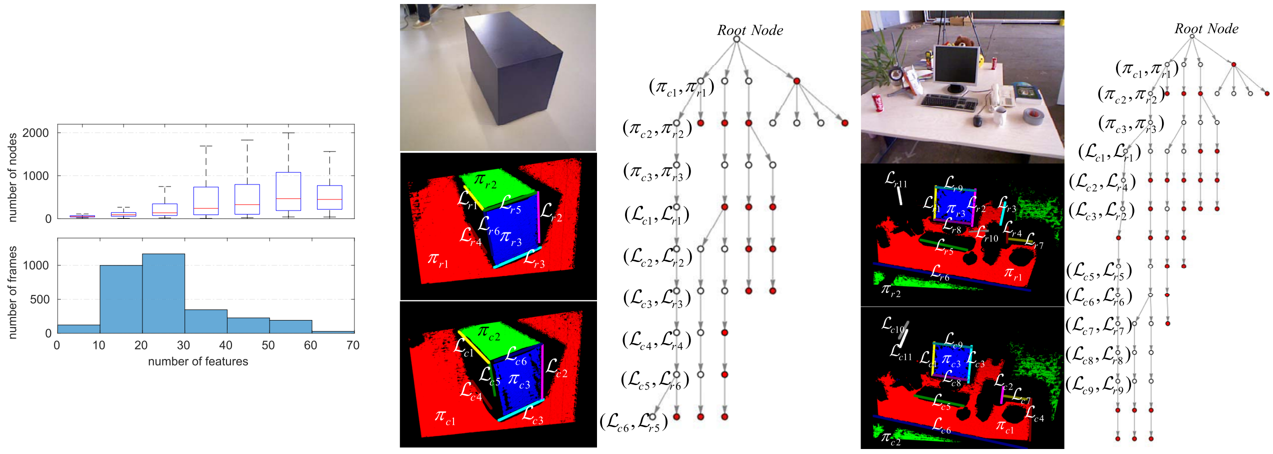
(a)

(b)

Fig. 4. Comparison with (a) LBD-based and (b) RANSAC-based line matching, respectively, in terms of (top) precision rate, (middle) recall rate and (bottom) runtime, respectively.

Then, the comparison with the RANSAC-based feature matching is conducted. The RANSAC algorithm is extensively used in the association of geometric features [12], [14]. In this experiment, the RANSAC algorithm is implemented according to the work of [12] and both planes and lines are used in the matching process. The precision and recall rates are presented in Fig. 4(b). For the RANSAC-based feature matching, the iteration proceeds until the association result with the most consistent inliers is found. Therefore, the RANSAC-based matching also presents high precision and recall rates. However, it is much more time-consuming than the IT-based method, as shown in the bottom figure of Fig. 4(b). Because the RANSAC algorithm is based on a random sampling process and it usually takes much time to converge to a good result. Differently, in the proposed IT-based method, the incorrect interpretations are naturally discarded during the tree expansion. A significant characteristic of the IT-based method is that feature association is accompanied by expansion of the IT in a closed form and no iterative process is involved, which makes our method quite time-saving.

In addition, the histogram at the bottom of Fig. 5(a) gives the number of frames (for nine image sequences used in Fig. 4) as a function of the quantity of extracted features (planes and lines) per frame. And at the top of Fig. 5(a), the statistics of the number of nodes in an IT corresponding to each bin of the histogram are shown in box plots. As can be seen in the histogram, the quantity of extracted features per frame is mainly distributed in the range of 10–30, and the corresponding quantity of nodes in the IT is fairly small.


number of nodes
2000
1000
0
number of frames
1000
500
0
0 10 20 30 40 50 60 70
number of features
Root Node
(π_{c1},π_{r1})
(π_{c2},π_{r2})
(π_{c3},π_{r3})
$(\mathcal{L}_{c1},\mathcal{L}_{r1})$
$(\mathcal{L}_{c2},\mathcal{L}_{r2})$
$(\mathcal{L}_{c3},\mathcal{L}_{r3})$
$(\mathcal{L}_{c4},\mathcal{L}_{r4})$
$(\mathcal{L}_{c5},\mathcal{L}_{r6})$
$(\mathcal{L}_{c6},\mathcal{L}_{r5})$
Root Node
(π_{c1},π_{r1})
(π_{c2},π_{r2})
(π_{c3},π_{r3})
$(\mathcal{L}_{c1},\mathcal{L}_{r1})$
$(\mathcal{L}_{c2},\mathcal{L}_{r4})$
$(\mathcal{L}_{c3},\mathcal{L}_{r2})$
$(\mathcal{L}_{c5},\mathcal{L}_{r5})$
$(\mathcal{L}_{c6},\mathcal{L}_{r6})$
$(\mathcal{L}_{c7},\mathcal{L}_{r7})$
$(\mathcal{L}_{c8},\mathcal{L}_{r8})$
$(\mathcal{L}_{c9},\mathcal{L}_{r9})$

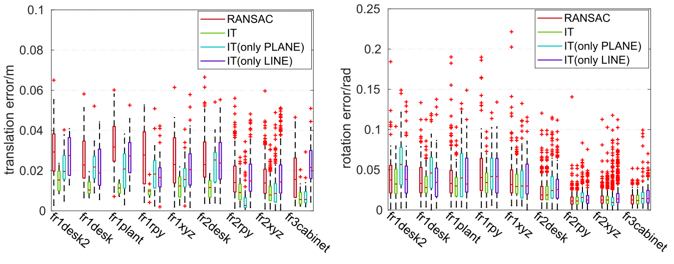
(a) (b) (c)

Fig. 5. (a) Bottom: histogram of the quantity of features (planes and lines). Top: statistics of the number of nodes in an IT corresponding to the features in each bin of the histogram. (b) and (c) (Left) Extracted plane and line features and (right) corresponding constructed IT structures for two scenes from the image sequences (b) fr3/cabinet and (c) fr2/desk, respectively. The null nodes are colored red in the IT. The nodes corresponding to the final association result are labeled by the indices of feature pairs.

Furthermore, though the number of nodes increases along with the quantity of features, the computational complexity remains tractable and the process of IT construction is proven to be time-saving, as shown in Fig. 4. Because through incrementally constraining the camera pose while expanding the tree, the feature association and pose estimation are jointly solved and the inconsistent hypotheses can be pruned timely in the IT expansion process. As a result, the search space of an IT is largely reduced, which leads to the high computational efficiency of the IT-based algorithm. Also, we visualize the constructed tree structures as well as the extracted features for two indoor scenes in Fig. 5(b) and (c), respectively. Only the nodes in the interpretation which is chosen as the association result are labeled by the corresponding indices of features for compactness. From the visualized tree structure we can see intuitively that most subtrees of an IT are pruned at the early stage of tree construction.

B. Experiments on Frame-to-Frame Registration

Unlike the visual descriptor-based feature matching, both the IT-based method and RANSAC-based method compute the camera transformation during the feature association. We run the IT-based and the RANSAC-based methods, respectively, using both plane and line features. Furthermore, the IT-based method is also run using only planes and only lines, respectively, to test the improvement on accuracy after combining them. The translation and rotation errors of the four methods are compared. As can be seen obviously from Fig. 6(a), the IT-based algorithm using both planes and lines gains the best results in term of the translation error in most cases. The rotation errors of four methods are shown in Fig. 6(b). Because the comparison results in terms of the rotation error cannot be clearly seen in the figure, we further compute the mean and variance of the rotation errors for each method as in Table III. It can be seen that the IT-based method using both features has the smallest mean and variance compared with the other three methods.



(a) translation error (b) rotation error

Fig. 6. Comparison of the matching error. (a) Translation error. (b) Rotation error.

TABLE III
COMPARISON OF THE ROTATION ERROR AMONG FOUR METHODS

	Mean(rad)			
	RANSAC	IT	IT(PLANE)	IT(LINE)
fr1/desk2	0.042	**0.036**	0.057	0.043
fr1/desk	0.039	**0.034**	0.048	0.038
fr1/plant	0.041	**0.031**	0.045	0.042
fr1/rpy	0.050	**0.036**	0.048	0.044
fr1/xyz	0.041	**0.033**	0.036	0.039
fr2/desk	**0.023**	**0.023**	0.030	0.029
fr2/rpy	0.013	**0.012**	0.017	0.014
fr2/xyz	0.014	**0.013**	**0.013**	0.017
fr3/cabinet	0.014	**0.013**	0.018	0.018
	Variance($\times 10^{-3} rad$)			
	RANSAC	IT	IT(PLANE)	IT(LINE)
fr1/desk2	0.782	**0.413**	1.048	0.851
fr1/desk	0.630	**0.387**	0.809	0.806
fr1/plant	1.056	**0.341**	0.979	0.829
fr1/rpy	1.485	**0.365**	0.725	0.813
fr1/xyz	1.104	**0.358**	0.639	0.677
fr2/desk	0.272	**0.231**	0.453	0.374
fr2/rpy	0.127	**0.059**	0.126	0.093
fr2/xyz	0.119	**0.080**	0.163	0.197
fr3/cabinet	**0.083**	**0.083**	0.342	0.198

For the pose estimation using only plane features, degeneracy frequently occurs such that the solution cannot be fully constrained. The issue of degeneracy was discussed in our

TABLE IV
RATIO OF CONSTRAINED CASES FOR THE IT-BASED METHOD USING ONLY PLANE FEATURES

	fr1/desk2	fr1/desk	fr1/plant	fr1/rpy	fr1/xyz
ratio	71.7%	79.8%	58.6%	70.1%	76.3%
	fr2/desk	fr2/rpy	fr2/xyz	fr3/cabinet	
ratio	64.8%	61.0%	79.0%	82.8%	

previous work [49] and was also mentioned in [12] and [20]. The estimation results of the IT-based method using only planes, which are shown in Fig. 6 does not include the degenerate cases. The ratio of cases that the planes provide sufficient constraints is given in Table IV, which shows that the degenerate cases occur in all scenes. Theoretically, the pose estimation using line features may also suffer from degeneracy. However, the quantity of extracted lines is generally larger than that of planes and only two non-planar lines can fully constrain the camera pose. As a result, degeneracy is much less likely to occur for line features. In our experiments, no degeneracy occurs over all the sequences. Nevertheless, Fig. 6 shows that the pose estimation using lines is less accurate than that using planes because lines are more likely to be detected on edges of objects, where the measurement noise is more severe [11]. Therefore, the combination of plane and line features not only increases the accuracy of pose estimation but also alleviates the problem of degeneracy.

C. Experiment on Joint Optimization

In this section, an ablation experiment is carried out to demonstrate the performance of the hybrid feature joint optimization. In the experiment, the IT-HYFAO-VO is run with the optimization using hybrid features, only plane features, and only line features, respectively, given the same feature association results. The root mean square errors (RMSEs) of the relative pose error (RPE) are computed for the three VO systems and the results are shown in Table V. It can be seen that the accuracy of the VO using hybrid features is obviously superior to the ones using only plane or line features. As is illustrated in Section V-B, the plane feature-based VO may suffer from the degenerate problems, and the ratios of nondegenerate cases for each sequence are also given in Table V. Note that the pose estimates corresponding to the degenerate solutions are not used to compute the RPE results. We can find that, even in the situations that the pose estimation can be fully constrained by the plane features, most of the optimization results using only plane features are less accurate than using hybrid features. As for the line features, although no degeneracy occurs, the resultant RPEs are larger than both the plane feature-based VO and the hybrid feature-based VO.

D. Evaluation of VO

In this subsection, the proposed IT-HYFAO-VO is evaluated. Three state-of-the-art VO algorithms are chosen for comparison: Prob-RGBD-VO [11], Canny-VO [50], and STING-VO that is presented in our previous work [49]. Prob-RGBD-VO is a robust VO that combines point, line, and plane features extracted through an RGB-D camera. The probabilistic plane and line fitting methods are used to model the uncertainties. Then, the pose is calculated considering the uncertainties of features. The STING-VO is achieved by aligning the plane features extracted from two successive frames. When the planes cannot fully constrain the problem, a scan alignment based on the statistical information grid is performed to estimate the remaining DoFs of the camera pose. Both the Prob-RGBD-VO and STING-VO use high-level geometric features to estimate the camera poses. The Canny-VO is an efficient RGB-D odometry system achieved by aligning the Canny edges extracted from the images. Though the Canny-VO does not use the high-level features directly, it exploits the geometric property of the edge structure during the process of a free-form curve registration.

Table VI presents the comparison results in terms of the RMSEs of the absolute trajectory error (ATE) and the RPE, among which the results of the Prob-RGBD-VO and the Canny-VO are reported in [11] and [50], respectively. It can be seen from Table VI that the proposed IT-HYFAO-VO performs better than or on par with the state-of-the-art VO algorithms on most sequences. Though the Prob-RGBD-VO combines multiple geometric features and considers the uncertainties of features as the IT-HYFAO-VO does, it does not exploit the geometrical relationships between different features. As for the edge-based VO algorithm, the measurement noises on the edges of objects are generally more severe than those on the plat surfaces. Therefore, the Canny-VO tends to get good results only when the edges can be clearly extracted. To further demonstrate the superior performance of the IT-HYFAO-VO pipeline, the overall statistical results are presented. Five standard statistics (mean, median, std., min and max) over all the sequences in Table VI are computed and listed in Table VII. As can be seen from Table VII, the performance of the IT-HYFAO-VO has overall better statistics than the comparison methods.

In addition, to demonstrate the real-time performance of the proposed VO pipeline, the statistics of runtime for the VO system are presented in Fig. 7(a). It can be seen that the proposed IT-HYFAO-VO runs at 7-10 Hz, which definitely meet general requirements for real-time performance. Furthermore, the module-wise average runtime is computed on each image sequence and the results are given in Fig. 7(b). It is obvious that the hybrid feature association and joint optimization modules, which are our main contributions, are quite time-saving, compared with the feature extraction procedure.

E. Evaluation of SLAM

To further verify the accuracy of the proposed IT-HYFAO-VO, we add a back-end factor-graph optimization process to yield a complete SLAM system (abbreviated as IT-HYFAO-SLAM in the following) and compare it with state-of-the-art RGB-D SLAM systems. As are listed in Table VIII, the comparison methods include point-feature-based SLAM system [51], geometric-feature-based ones [49], [52], multifeature-based ones [53], [54], and

TABLE V
RESULTS OF USING DIFFERENT FEATURES IN JOINT OPTIMIZATION IN TERMS OF RMSE OF RPE

	IT-HYFAO-VO /hybrid features	IT-HYFAO-VO /only line feature	IT-HYFAO-VO /only plane feature (ratio of constrained cases)
fr1/desk	**0.025m;1.7deg**	0.045m;2.4deg	0.033m;1.9deg (79.8%)
fr1/xyz	**0.012m;0.7deg**	0.023m;1.7deg	0.031m;1.9deg (76.3%)
fr2/xyz	**0.004m;0.3deg**	0.023m;1.7deg	0.031m;1.9deg (79.0%)
fr2/desk	**0.007m;0.6deg**	0.023m;1.7deg	0.031m;1.9deg (64.8%)
fr1/360	**0.062m;2.1deg**	0.104m;2.8deg	0.112m;3.0deg (68.9%)
fr3/str_notex_far	0.014m;**0.5deg**	0.016m;0.8deg	**0.011m;0.5deg** (59.1%)
fr3/cabinet	**0.016m;1.3deg**	0.021m;1.7deg	**0.016m**;1.5deg (82.8%)
fr3/office	**0.010m;0.6deg**	0.044m;3.6deg	0.037m;1.9deg (88.3%)
lr0	**0.011m;0.3deg**	0.014m;0.5deg	0.019m;1.1deg (73.6%)

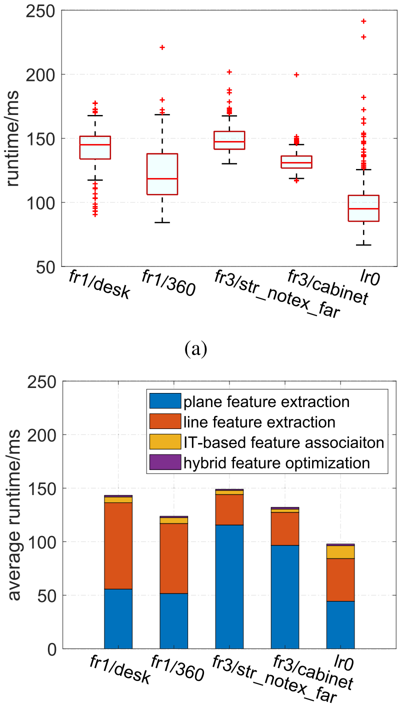
TABLE VI
COMPARISON OF VO IN TERMS OF RMSEs OF ATE AND RPE

	ATE				RPE			
	IT-HYFAO-VO	Prob-RGBD-VO	Canny-VO	STING-VO	IT-HYFAO-VO	Prob-RGBD-VO	Canny-VO	STING-VO
fr1/desk	**0.040m**	**0.040m**	0.044m	0.041m	0.025m;**1.7deg**	**0.023m;1.7deg**	0.031m;1.9deg	0.025m;1.9deg
fr1/360	**0.088m**	0.091m	0.242m	0.122m	**0.062m;2.1deg**	0.064m;2.7deg	0.121m;4.0deg	0.092m;3.1deg
fr3/str_notex_far	**0.031m**	0.054m	**0.031m**	0.040m	**0.014m;0.5deg**	0.019m;0.7deg	0.027m;**0.5deg**	**0.014m**;0.8deg
fr3/cabinet	**0.051m**	0.200m	0.057m	0.070m	0.016m;1.3deg	0.039m;1.8deg	0.036m;1.6deg	**0.011m;1.0deg**
lr0	0.058m	0.059m	**0.035m**	0.071m	0.011m;**0.3deg**	**0.006m**;0.5deg	0.014m;0.6deg	0.019m;0.7deg

TABLE VII
OVERALL STATISTICS OF THE RESULTS IN TABLE VI

		IT-HYFAO-VO	Prob-RGBD-VO	Canny-VO	STING-VO
ATE	Mean	**0.054m**	0.089m	0.082m	0.069m
	Median	0.051m	0.059m	**0.044m**	0.070m
	Std.	**0.022m**	0.065m	0.090m	0.033m
	Min	**0.031m**	0.040m	**0.031m**	0.040m
	Max	**0.088m**	0.200m	0.242m	0.122m
RPE	Mean	**0.025m;1.2deg**	0.030m;1.5deg	0.046m;1.7deg	0.032m;1.5deg
	Median	**0.016m**;1.3deg	0.023m;1.7deg	0.031m;1.6deg	0.019m;**1.0deg**
	Std.	**0.021m;0.7deg**	0.022m;0.9deg	0.042m;1.4deg	0.033m;1.0deg
	Min	0.011m;**0.3deg**	**0.006m**;0.5deg	0.014m;0.5deg	0.011m;0.7deg
	Max	**0.062m;2.1deg**	0.064m;2.7deg	0.121m;4.0deg	0.092m;3.1deg

map-fusion-based ones [55]–[57]. The ORB-SLAM2 [51] is widely acknowledged as the most efficient open-source implementation of the point-feature-based SLAM algorithm. Both the CPA-SLAM [52] and STING-SLAM [49] use plane features. The CPA-SLAM tracks the camera motion via a direct image alignment toward the keyframes as well as a global plane model, while the STING-SLAM computes the pose of the RGB-D camera directly using parameters of plane features. The PL-SLAM [53] and PinpointSLAM [54] combine the geometric features (planes or lines) with the point features. And the ElasticFusion [55], GC-SLAM [56] and PSM-SLAM [57] are based on map fusion and dense alignment of the scans. We run the IT-HYFAO-SLAM in various scenes from different datasets and compare it with the other eight methods, whose results were reported in the literature [52]–[57], as shown in Table IV. It is shown from Table VIII that the IT-HYFAO-SLAM system compares favorably with other state-of-the-art SLAM methods. As in the evaluation of the VO methods, the overall statistical results are also computed for the SLAM methods. Table IX lists the five statistics over all the sequences in Table VI. We can see that the IT-HYFAO-SLAM system get the best results in terms of the mean, std. and max statistics, which further demonstrates the good performance of the proposed method.



(a)

(b)

Fig. 7. (a) Statistics of the runtime and (b) modulewise average runtime for the IT-HYFAO-VO on the sequences in Table VI.

TABLE VIII

COMPARISON OF SLAM IN TERM OF ATE RMSE

	IT-HYFAO-SLAM	ORB-SLAM2	Elastic Fusion	GC-SLAM	PSM-SLAM	PL-SLAM	Pinpoint SLAM	CPA-SLAM	STING-SLAM
fr1/xyz	**0.011m**	0.013m	**0.011m**	**–**	**0.011m**	0.012m	0.015m	**0.011m**	**0.011m**
fr1/rpy	**0.020m**	0.025m	0.025m	–	0.021m	–	–	0.024m	0.025m
fr1/desk	**0.015m**	0.016m	0.020m	0.019m	0.016m	–	–	0.018m	0.030m
fr1/desk2	**0.020m**	0.022m	0.048m	–	0.026m	–	–	0.029m	0.037m
fr1/room	**0.045m**	0.047m	0.068m	–	0.052m	–	–	0.055m	0.083m
fr2/xyz	0.011m	**0.004m**	0.011m	0.011m	–	**0.004m**	0.012m	0.014m	0.010m
fr2/desk	0.043m	**0.009m**	0.071m	–	0.080m	–	0.063m	0.046m	0.053m
fr3/office	0.022m	**0.010m**	0.017m	0.026m	0.031m	0.019m	0.026m	0.025m	0.034m
fr3/nst_tex_near	**0.016m**	0.019m	**0.016m**	**0.016m**	–	0.020m	–	**0.016m**	0.018m
fr3/str_tex_far	**0.008m**	0.015m	0.013m	–	–	0.009m	0.026m	–	0.009m
fr3/str_ntex_near	**0.020m**	failed	0.021m	–	–	–	–	–	0.037m
fr3/str_ntex_far	**0.027m**	failed	0.030m	–	–	–	–	–	0.060m
kt0	0.009m	0.019m	0.009m	0.006m	0.005m	–	**0.004m**	–	0.011m
kt1	**0.006m**	0.058m	0.009m	**0.006m**	0.008m	–	0.019m	–	**0.006m**
kt2	**0.008m**	0.047m	0.014m	**0.008m**	0.054m	–	**0.008m**	0.089m	0.021m
kt3	**0.008m**	0.037m	0.106m	0.010m	0.030m	–	0.016m	0.009m	0.015m

TABLE IX

OVERALL STATISTICS OF THE RESULTS IN TABLE VIII

	IT-HYFAO-SLAM	ORB-SLAM2	Elastic Fusion	GC-SLAM	PSM-SLAM	PL-SLAM	Pinpoint SLAM	CPA-SLAM	STING-SLAM
Mean	**0.018m**	0.022m	0.031m	0.020m	0.031m	0.028m	0.028m	0.032m	0.029m
Median	0.016m	0.018m	0.019m	**0.014m**	0.024m	0.017m	0.018m	0.021m	0.023m
Std.	**0.012m**	0.016m	0.028m	0.024m	0.029m	0.029m	0.027m	0.029m	0.022m
Min	0.006m	**0.004m**	0.009m	0.006m	0.005m	**0.004m**	**0.004m**	0.009m	0.006m
Max	**0.045m**	0.058m	0.106m	0.106m	0.106m	0.106m	0.106m	0.106m	0.083m

VI. CONCLUSION

In this article, the IT-HYFAO-VO system has been developed using high-level geometric features (planes and lines). To associate multiple geometric features simultaneously, a unified framework has been proposed based on the IT. The IT expansion method has been elaborately designed for the association of multiple geometric features. The proposed method has been evaluated and compared with the state-of-the-art methods in different public datasets and has shown good performance.

In the proposed IT-HYFAO-VO method, complementary advantages of different features are exploited to improve the accuracy and alleviates the problem of degeneracy. Compared with the widely used NN search-based or RANSAC-based association of geometric features, all the possible hypotheses are properly structured in an IT structure and optimal solutions to both the feature association and the pose estimation can be obtained in a closed form, which guarantees that the associated features are consistent under common transformations. Additionally, the proposed framework is theoretically extensible to any combination of different types of features, such as points, lines, and planes. Furthermore, the geometric features, such as planes and lines, encode more higher-level semantic information of indoor environments, which is beneficial for robot tasks like scene recognition and understanding. It needs to be pointed out that the proposed IT-HYFAO-VO method is theoretically independent of the sensor type as well as the specific feature extraction algorithm. Therefore, IT-HYFAO-VO can be easily extended to other sensors, such as a monocular vision or Lidar sensor, as long as parameters of the extracted features can be computed. In future works, we plan to combine the point features with high-level geometric features in one unified framework to further improve the accuracy and robustness of the system. Further, the visual appearance provided by the point features as well as the structural information provided by the geometric features is planned to be exploited simultaneously to benefit the high-level tasks of the robot.

APPENDIX A
ROTATION CONSISTENCY

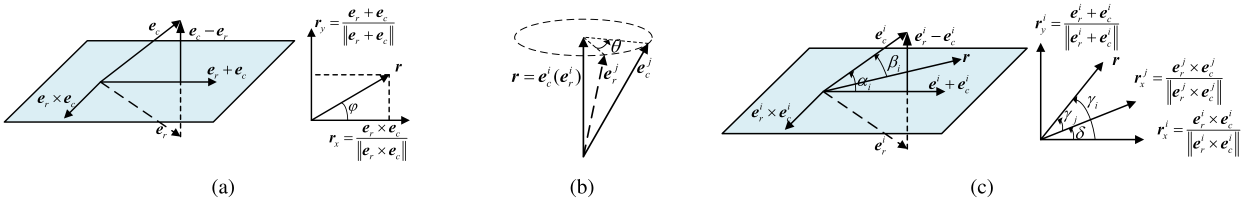
In this appendix, the detailed solution to (3) is given. The rigid rotation in 3-D space can be formulated as $\boldsymbol{e}_c = \boldsymbol{R}\boldsymbol{e}_r$, with $\boldsymbol{e}$ being the unit direction vector and $\boldsymbol{R} \in \mathbb{SO}(3)$ the rotation matrix. Specifically, $\boldsymbol{R}$ can be expressed as

$$\boldsymbol{R} = \text{Rot}(\boldsymbol{r}, \theta) = \cos\theta \boldsymbol{I} + (1 - \cos\theta)\boldsymbol{r}\boldsymbol{r}^T + \sin\theta[\boldsymbol{r}]_\times \quad (16)$$

where $\boldsymbol{r}$ is the unit axis about which the rotation takes place, and θ is the angle of rotation about $\boldsymbol{r}$. Given the rotation axis $\boldsymbol{r}$ and a pair of feature directions $\boldsymbol{e}_c$ and $\boldsymbol{e}_r$ ($\boldsymbol{r}$, $\boldsymbol{e}_c$ and $\boldsymbol{e}_r$ are not collinear), substitute (16) into $\boldsymbol{e}_c = \boldsymbol{R}\boldsymbol{e}_r$ and the rotation angle θ can be calculated by

$$\theta = \Theta(\boldsymbol{r}, \boldsymbol{e}_c, \boldsymbol{e}_r) = \text{atan2}(\sin\theta, \cos\theta) \quad (17)$$

$$\cos\theta = 1 - \frac{1 - \boldsymbol{e}_r^T\boldsymbol{e}_c}{1 - \left(\boldsymbol{r}^T\boldsymbol{e}_c\right)\left(\boldsymbol{r}^T\boldsymbol{e}_r\right)}, \quad \sin\theta = \frac{\left(\boldsymbol{r} \times \boldsymbol{e}_r\right)^T\boldsymbol{e}_c}{1 - \left(\boldsymbol{r}^T\boldsymbol{e}_c\right)\left(\boldsymbol{r}^T\boldsymbol{e}_r\right)}. \quad (18)$$


$\boldsymbol{e}_c$
$\boldsymbol{e}_c - \boldsymbol{e}_r$
$\boldsymbol{e}_r + \boldsymbol{e}_c$
$\boldsymbol{e}_r \times \boldsymbol{e}_c$
$\boldsymbol{e}_r$
$\boldsymbol{r}_y = \frac{\boldsymbol{e}_r + \boldsymbol{e}_c}{\|\boldsymbol{e}_r + \boldsymbol{e}_c\|}$
$\boldsymbol{r}$
φ
$\boldsymbol{r}_x = \frac{\boldsymbol{e}_r \times \boldsymbol{e}_c}{\|\boldsymbol{e}_r \times \boldsymbol{e}_c\|}$
θ
$\boldsymbol{r} = \boldsymbol{e}_c^i(\boldsymbol{e}_r^i)$
$\boldsymbol{e}_r^j$
$\boldsymbol{e}_c^j$
$\boldsymbol{e}_c^i$
$\boldsymbol{e}_r^i - \boldsymbol{e}_c^i$
β_i
$\boldsymbol{r}$
α_i
$\boldsymbol{e}_r^i + \boldsymbol{e}_c^i$
$\boldsymbol{e}_r^i \times \boldsymbol{e}_c^i$
$\boldsymbol{e}_r^i$
$\boldsymbol{r}_y^i = \frac{\boldsymbol{e}_r^i + \boldsymbol{e}_c^i}{\|\boldsymbol{e}_r^i + \boldsymbol{e}_c^i\|}$
$\boldsymbol{r}$
$\boldsymbol{r}_x^j = \frac{\boldsymbol{e}_r^j \times \boldsymbol{e}_c^j}{\|\boldsymbol{e}_r^j \times \boldsymbol{e}_c^j\|}$
γ_i
γ_j
δ
$\boldsymbol{r}_x^i = \frac{\boldsymbol{e}_r^i \times \boldsymbol{e}_c^i}{\|\boldsymbol{e}_r^i \times \boldsymbol{e}_c^i\|}$

(a) (b) (c)

Fig. 8. (a) Left: rotation axes lie on the plane perpendicular to $\boldsymbol{e}_c - \boldsymbol{e}_r$. Right: Resultant rotation axis with 1DoF can be expressed by two basis vectors $\boldsymbol{r}_x$ and $\boldsymbol{r}_y$ on the plane. (b) If $\boldsymbol{e}_c^i = \boldsymbol{e}_r^i$ and $\boldsymbol{e}_c^j \neq \boldsymbol{e}_r^j$, the rotation axis is simply $\boldsymbol{e}_c^i(\boldsymbol{e}_r^i)$. (c) Left: illustration of the angles α_i and β_i. Right: illustration of the angles γ_i, γ_j and δ. (a) Case I. (b) Case II. (c) Case III.

Because a rigid rotation preserves orientation, $\langle \boldsymbol{e}_c^i, \boldsymbol{e}_c^j \rangle = \langle \boldsymbol{e}_r^i, \boldsymbol{e}_r^j \rangle$ holds true if there exists a feasible solution $\boldsymbol{R}$ to (3), with $\langle \cdot, \cdot \rangle$ denoting the angle between two vectors. Besides, a rotation also preserves the angle between the transformed vector and the direction of rotation. As a result, the set of potential rotation axes that satisfies (3) is given by

$$\left\{\boldsymbol{r} \middle| \boldsymbol{r}^T \boldsymbol{e}_c^i = \boldsymbol{r}^T \boldsymbol{e}_r^i, \boldsymbol{r}^T \boldsymbol{e}_c^j = \boldsymbol{r}^T \boldsymbol{e}_r^j, \boldsymbol{r}^T \boldsymbol{r} = 1\right\}. \tag{19}$$

According to the spatial configuration of the direction vectors, the solution to (3) can be classified into four cases, which are presented in the following.

Case I: $\boldsymbol{e}_c^i = \boldsymbol{e}_c^j$ and $\boldsymbol{e}_r^i = \boldsymbol{e}_r^j$.

In this case, the direction vectors of the two features $\boldsymbol{e}_c^i$ and $\boldsymbol{e}_c^j$ (or $\boldsymbol{e}_r^i$ and $\boldsymbol{e}_r^j$) in the same coordinate system coincide with each other. Let $\boldsymbol{e}_c = \boldsymbol{e}_c^i = \boldsymbol{e}_c^j$ and $\boldsymbol{e}_r = \boldsymbol{e}_r^i = \boldsymbol{e}_r^j$.

If $\boldsymbol{e}_c = \boldsymbol{e}_r$, then the rotation axis is $\boldsymbol{e}_c$ and the set of consistent rotations is $\mathcal{R} = \{\boldsymbol{R} | \boldsymbol{R} = \mathrm{Rot}(\boldsymbol{e}_c, \varphi), \varphi \in \mathbb{R}\}$. If $\boldsymbol{e}_c \neq \boldsymbol{e}_r$, the set of rotation axes defined in (19) can be rewritten as

$$\left\{\boldsymbol{r} \middle| \boldsymbol{r}^T (\boldsymbol{e}_c - \boldsymbol{e}_r) = 0, \boldsymbol{r}^T \boldsymbol{r} = 1\right\}. \tag{20}$$

It is clear from (20) that the unit axis $\boldsymbol{r}$ lies on the plane perpendicular to vector $\boldsymbol{e}_c - \boldsymbol{e}_r$, as seen in Fig. 8(a). Thus, $\boldsymbol{r}$ has one degree of freedom (DoF). Without loss of generality, let

$$\boldsymbol{r}_x = \frac{\boldsymbol{e}_r \times \boldsymbol{e}_c}{\|\boldsymbol{e}_r \times \boldsymbol{e}_c\|}, \quad \boldsymbol{r}_y = \frac{\boldsymbol{e}_r + \boldsymbol{e}_c}{\|\boldsymbol{e}_r + \boldsymbol{e}_c\|} \tag{21}$$

which form a basis on the plane perpendicular to $\boldsymbol{e}_c - \boldsymbol{e}_r$. The rotation axis satisfying (20) can be represented by

$$\boldsymbol{r}(\varphi) = \boldsymbol{r}_x \cos\varphi + \boldsymbol{r}_y \sin\varphi, \quad \varphi \in \mathbb{R} \tag{22}$$

$\forall \varphi \in \mathbb{R}$, the angle of rotation corresponding to $\boldsymbol{r}(\varphi)$ can be computed by $\theta(\varphi) = \Theta(\boldsymbol{r}(\varphi), \boldsymbol{e}_c, \boldsymbol{e}_))$.

In conclusion, for the case that $\boldsymbol{e}_c^i = \boldsymbol{e}_c^j$ and $\boldsymbol{e}_r^i = \boldsymbol{e}_r^j$, the rigid rotation that satisfies the rotation consistency has one DoF. The set of consistent rotations is denoted by $\mathcal{R} = \{\boldsymbol{R} | \boldsymbol{R} = R(\varphi), \varphi \in \mathbb{R}\}$, with

$$R(\varphi) = \begin{cases} \mathrm{Rot}(\boldsymbol{e}_c, \varphi), & \text{if } \boldsymbol{e}_c = \boldsymbol{e}_r \\ \mathrm{Rot}(\boldsymbol{r}(\varphi), \theta(\varphi)), & \text{if } \boldsymbol{e}_c \neq \boldsymbol{e}_r. \end{cases} \tag{23}$$

Case II: At least one of the equations $\boldsymbol{e}_c^i = \boldsymbol{e}_r^i$ and $\boldsymbol{e}_c^j = \boldsymbol{e}_r^j$ is satisfied ($\boldsymbol{e}_c^i \neq \boldsymbol{e}_c^j$).

In this case, the directions of at least one pair of corresponding geometric features coincide with each other.

If both $\boldsymbol{e}_c^i = \boldsymbol{e}_r^i$ and $\boldsymbol{e}_c^j = \boldsymbol{e}_r^j$ hold true, then the resultant rotation matrix is $\boldsymbol{R} = \boldsymbol{I}$. If $\boldsymbol{e}_c^i = \boldsymbol{e}_r^i$ and $\boldsymbol{e}_c^j \neq \boldsymbol{e}_r^j$, then the rotation axis is $\boldsymbol{r} = \boldsymbol{e}_c^i$ and the rotation angle can be computed by $\theta = \Theta(\boldsymbol{r}, \boldsymbol{e}_c^j, \boldsymbol{e}_r^j)$, as illustrated in Fig. 8(b). Likewise, if $\boldsymbol{e}_c^j = \boldsymbol{e}_r^j$ and $\boldsymbol{e}_c^i \neq \boldsymbol{e}_r^i$, then $\boldsymbol{r} = \boldsymbol{e}_c^j$, $\theta = \Theta(\boldsymbol{r}, \boldsymbol{e}_c^i, \boldsymbol{e}_r^i)$. In summary, the resultant rotation matrix can be determined by

$$\boldsymbol{R} = \begin{cases} \boldsymbol{I}, & \text{if } \boldsymbol{e}_c^i = \boldsymbol{e}_r^i \text{ and } \boldsymbol{e}_c^j = \boldsymbol{e}_r^j \\ \mathrm{Rot}\left(\boldsymbol{e}_c^i, \Theta\left(\boldsymbol{e}_c^i, \boldsymbol{e}_c^j, \boldsymbol{e}_r^j\right)\right), & \text{if } \boldsymbol{e}_c^i = \boldsymbol{e}_r^i \text{ and } \boldsymbol{e}_c^j \neq \boldsymbol{e}_r^j \\ \mathrm{Rot}\left(\boldsymbol{e}_c^j, \Theta\left(\boldsymbol{e}_c^j, \boldsymbol{e}_c^i, \boldsymbol{e}_r^i\right)\right), & \text{if } \boldsymbol{e}_c^i \neq \boldsymbol{e}_r^i \text{ and } \boldsymbol{e}_c^j = \boldsymbol{e}_r^j. \end{cases} \tag{24}$$

Case III: $(\boldsymbol{e}_c^i - \boldsymbol{e}_r^i) \times (\boldsymbol{e}_c^j - \boldsymbol{e}_r^j) = 0$ ($\boldsymbol{e}_c^i \neq \boldsymbol{e}_c^j$, $\boldsymbol{e}_c^i \neq \boldsymbol{e}_r^i$, $\boldsymbol{e}_c^j \neq \boldsymbol{e}_r^j$).

In this case, although the rotation axis $\boldsymbol{r}$ cannot be determined directly by (19), it can be calculated by solving

$$\Theta\left(\boldsymbol{r}, \boldsymbol{e}_c^i, \boldsymbol{e}_r^i\right) = \Theta\left(\boldsymbol{r}, \boldsymbol{e}_c^j, \boldsymbol{e}_r^j\right). \tag{25}$$

From (17) and (18), (25) can be rewritten as

$$\begin{cases} 1 - \dfrac{1 - {\boldsymbol{e}_r^i}^T \boldsymbol{e}_c^i}{1 - \left(\boldsymbol{r}^T \boldsymbol{e}_c^i\right)\left(\boldsymbol{r}^T \boldsymbol{e}_r^i\right)} = 1 - \dfrac{1 - {\boldsymbol{e}_r^j}^T \boldsymbol{e}_c^j}{1 - \left(\boldsymbol{r}^T \boldsymbol{e}_c^j\right)\left(\boldsymbol{r}^T \boldsymbol{e}_r^j\right)} \\ \dfrac{\left(\boldsymbol{r} \times \boldsymbol{e}_r^i\right)^T \boldsymbol{e}_c^i}{1 - \left(\boldsymbol{r}^T \boldsymbol{e}_c^i\right)\left(\boldsymbol{r}^T \boldsymbol{e}_r^i\right)} = \dfrac{\left(\boldsymbol{r} \times \boldsymbol{e}_r^j\right)^T \boldsymbol{e}_c^j}{1 - \left(\boldsymbol{r}^T \boldsymbol{e}_c^j\right)\left(\boldsymbol{r}^T \boldsymbol{e}_r^j\right)}. \end{cases} \tag{26}$$

Denote several included angles by

$$\begin{aligned} \alpha_i &= \frac{1}{2}\left\langle \boldsymbol{e}_r^i, \boldsymbol{e}_c^i \right\rangle, \quad \alpha_j = \frac{1}{2}\left\langle \boldsymbol{e}_r^j, \boldsymbol{e}_c^j \right\rangle \\ \beta_i &= \left\langle \boldsymbol{r}, \boldsymbol{e}_r^i \right\rangle = \left\langle \boldsymbol{r}, \boldsymbol{e}_c^i \right\rangle, \quad \beta_j = \left\langle \boldsymbol{r}, \boldsymbol{e}_r^j \right\rangle = \left\langle \boldsymbol{r}, \boldsymbol{e}_c^j \right\rangle \\ \gamma_i &= \left\langle \boldsymbol{r}, \boldsymbol{e}_r^i \times \boldsymbol{e}_c^i \right\rangle, \quad \gamma_j = \left\langle \boldsymbol{r}, \boldsymbol{e}_r^j \times \boldsymbol{e}_c^j \right\rangle. \end{aligned} \tag{27}$$

After some algebraic calculation, (26) can be rewritten as

$$\frac{\sin^2 \alpha_i}{\sin^2 \beta_i} = \frac{\sin^2 \alpha_j}{\sin^2 \beta_j} \tag{28}$$

$$\frac{\cos\gamma_i \sin\alpha_i \cos\alpha_i}{\sin^2 \beta_i} = \frac{\cos\gamma_j \sin\alpha_j \cos\alpha_j}{\sin^2 \beta_j}. \tag{29}$$

From the criteria $\boldsymbol{e}_c^i \neq \boldsymbol{e}_r^i$ and $\boldsymbol{e}_c^j \neq \boldsymbol{e}_r^j$, it is obvious that $\sin\alpha_i \neq 0$ and $\sin\alpha_j \neq 0$. And from $\boldsymbol{e}_c^i \neq \boldsymbol{e}_c^j$, we know that if $\cos\alpha_i = 0$ then $\cos\alpha_j \neq 0$, and vice versa. If $\cos\alpha_i = 0$, from (29) we know that $\cos\gamma_j = 0$ and the rotation axis is $\boldsymbol{r} = \eta(\boldsymbol{e}_c^j + \boldsymbol{e}_r^j)$ with η denoting the normalizing factor. Likewise, if $\cos\alpha_j = 0$, then $\boldsymbol{r} = \eta(\boldsymbol{e}_c^i + \boldsymbol{e}_r^i)$.

If $\cos\alpha_i \neq 0$ and $\cos\alpha_j \neq 0$, combining (28) and (29) yields

$$\frac{\cos\gamma_i}{\tan\alpha_i} = \frac{\cos\gamma_j}{\tan\alpha_j}. \tag{30}$$

Let

$$\boldsymbol{r}_x^i = \frac{\boldsymbol{e}_r^i \times \boldsymbol{e}_c^i}{\left\|\boldsymbol{e}_r^i \times \boldsymbol{e}_c^i\right\|}, \quad \boldsymbol{r}_y^i = \frac{\boldsymbol{e}_r^i + \boldsymbol{e}_c^i}{\left\|\boldsymbol{e}_r^i + \boldsymbol{e}_c^i\right\|}$$
$$\boldsymbol{r}_x^j = \frac{\boldsymbol{e}_r^j \times \boldsymbol{e}_c^j}{\left\|\boldsymbol{e}_r^j \times \boldsymbol{e}_c^j\right\|}, \quad \boldsymbol{r}_y^j = \frac{\boldsymbol{e}_r^j + \boldsymbol{e}_c^j}{\left\|\boldsymbol{e}_r^j + \boldsymbol{e}_c^j\right\|}. \tag{31}$$

It is obvious that $\boldsymbol{r}_x^i$ and $\boldsymbol{r}_y^i$ are orthogonal to each other. γ_i is the angle between $\boldsymbol{r}$ and $\boldsymbol{r}_x^i$, as illustrated in Fig. 8(c), and it can be expressed by

$$\gamma_i = \operatorname{atan2}\left(\boldsymbol{r}^T\boldsymbol{r}_y^i, \boldsymbol{r}^T\boldsymbol{r}_x^i\right). \tag{32}$$

And let

$$\delta = \operatorname{atan2}\left({\boldsymbol{r}_x^j}^T\boldsymbol{r}_y^i, {\boldsymbol{r}_x^j}^T\boldsymbol{r}_x^i\right). \tag{33}$$

From (27), (32) and (33), it can be known that $\gamma_j = \gamma_i - \delta$, as shown in Fig. 8(c). Substitute γ_j into (30) and we obtain

$$\gamma_i = \operatorname{atan2}\left(\frac{\tan\alpha_j}{\tan\alpha_i} - \cos\delta,\ \sin\delta\right). \tag{34}$$

Thus, the axis $\boldsymbol{r}$ is

$$\boldsymbol{r} = \boldsymbol{r}_x^i\cos\gamma_i + \boldsymbol{r}_y^i\sin\gamma_i. \tag{35}$$

Note that for the aforementioned special cases $\cos\alpha_i = 0$ and $\cos\alpha_j = 0$, the results can also be included in the unified formulation (35).

Case IV: The general case $[(\boldsymbol{e}_c^i - \boldsymbol{e}_r^i) \times (\boldsymbol{e}_c^j - \boldsymbol{e}_r^j) \neq 0, \boldsymbol{e}_c^i \neq \boldsymbol{e}_c^j, \boldsymbol{e}_c^i \neq \boldsymbol{e}_r^i, \boldsymbol{e}_c^j \neq \boldsymbol{e}_r^j]$.

According to (19), the rotation axis lies on the planes perpendicular to the vectors $\boldsymbol{e}_c^i - \boldsymbol{e}_r^i$ and $\boldsymbol{e}_c^j - \boldsymbol{e}_r^j$, respectively. Therefore, the rotation axis can be determined by $\boldsymbol{r} = \eta(\boldsymbol{e}_c^i - \boldsymbol{e}_r^i) \times (\boldsymbol{e}_c^j - \boldsymbol{e}_r^j)$, where η is the normalizing factor. Then, the rotation angles $\theta_i = \Theta(\boldsymbol{r}, \boldsymbol{e}_c^i, \boldsymbol{e}_r^i)$ and $\theta_j = \Theta(\boldsymbol{r}, \boldsymbol{e}_c^j, \boldsymbol{e}_r^j)$ are computed, respectively. If $\theta_i = \theta_j$, then the two nodes are rotation consistent under $\boldsymbol{R} = \operatorname{Rot}(\boldsymbol{r}, \theta_i)$.

Among the above four cases, the 3DoF rotation can be fully constrained in three cases, except case I, which corresponds to a special configuration of the directions of features. In the following, the translation consistency is calculated given the results of the rotation consistency.

Appendix B
Translation Consistency

In this section, the computation of the translation consistency is presented. Three different cases are dealt with separately, i.e., plane-plane case, line-line case and plane-line case.

A. Plane-Plane Case

$\mathcal{F}_c^i = \pi_c^i, \mathcal{F}_r^i = \pi_r^i, \mathcal{F}_c^j = \pi_c^j, \mathcal{F}_r^j = \pi_r^j$.

When two pairs of plane features are known to be rotation consistent under any $\boldsymbol{R} \in \mathcal{R}$, their consistent translations can be calculated by solving the following linear equations:

$$\begin{aligned} {\boldsymbol{n}_c^i}^T\boldsymbol{t} &= d_r^i - d_c^i \\ {\boldsymbol{n}_c^j}^T\boldsymbol{t} &= d_r^j - d_c^j. \end{aligned} \tag{36}$$

Rewriting (36) in a matrix form yields

$$\boldsymbol{A}_{\text{pp}}\boldsymbol{t} = \boldsymbol{b}_{\text{pp}} \tag{37}$$

$$\boldsymbol{A}_{\text{pp}} = \begin{bmatrix} {\boldsymbol{n}_c^i}^T \\ {\boldsymbol{n}_c^j}^T \end{bmatrix}, \quad \boldsymbol{b}_{\text{pp}} = \begin{bmatrix} d_r^i - d_c^i \\ d_r^j - d_c^j \end{bmatrix}. \tag{38}$$

If $\boldsymbol{n}_c^i = \boldsymbol{n}_c^j = \boldsymbol{n}_c$, i.e., the planes in the same coordinate systems are parallel to each other, it is obvious that $\operatorname{rank}(\boldsymbol{A}_{\text{pp}}) = 1$. In this case, if the linear system (37) has solutions, it requires that $\operatorname{rank}([\boldsymbol{A}_{\text{pp}}|\boldsymbol{b}_{\text{pp}}]) = 1$, i.e., $d_r^i - d_c^i = d_r^j - d_c^j$. Since $\operatorname{rank}([\boldsymbol{A}_{\text{pp}}|\boldsymbol{b}_{\text{pp}}]) = \operatorname{rank}(\boldsymbol{A}_{\text{pp}}) < 3$, the linear system (37) has infinite solutions with two DoFs

$$\boldsymbol{t} = \boldsymbol{t}_{\text{pp1}} + [\boldsymbol{w}_{\text{pp1}}]_\times\boldsymbol{\mu} \tag{39}$$

where

$$\boldsymbol{w}_{\text{pp1}} = \boldsymbol{n}_c, \ \boldsymbol{\mu} \in \mathbb{R}^3$$
$$\boldsymbol{t}_{\text{pp1}} = \left(\boldsymbol{A}_{\text{pp1}}^T\boldsymbol{A}_{\text{pp1}}\right)^{-1}\boldsymbol{A}_{\text{pp1}}^T\boldsymbol{b}_{\text{pp1}}$$
$$\boldsymbol{A}_{\text{pp1}} = \begin{bmatrix} \boldsymbol{A}_{\text{pp}} \\ [\boldsymbol{w}_{\text{pp1}}]_\times \end{bmatrix}, \quad \boldsymbol{b}_{\text{pp1}} = \begin{bmatrix} \boldsymbol{b}_{\text{pp}} \\ \boldsymbol{0} \end{bmatrix}. \tag{40}$$

If $\boldsymbol{n}_c^i \neq \boldsymbol{n}_c^j$, i.e., the planes in the same coordinate system are nonparallel, then $\operatorname{rank}(\boldsymbol{A}_{\text{pp}}) = 2$. Because $\operatorname{rank}([\boldsymbol{A}_{\text{pp}}|\boldsymbol{b}_{\text{pp}}]) = 2$ holds true, the linear system (37) has infinite solutions which have one DoF

$$\boldsymbol{t} = \boldsymbol{t}_{\text{pp2}} + \mu\boldsymbol{w}_{\text{pp2}} \tag{41}$$

where

$$\boldsymbol{w}_{\text{pp2}} = \boldsymbol{n}_c^i \times \boldsymbol{n}_c^j, \quad \mu \in \mathbb{R}$$
$$\boldsymbol{t}_{\text{pp2}} = \left(\boldsymbol{A}_{\text{pp2}}^T\boldsymbol{A}_{\text{pp2}}\right)^{-1}\boldsymbol{A}_{\text{pp2}}^T\boldsymbol{b}_{\text{pp2}}$$
$$\boldsymbol{A}_{\text{pp2}} = \begin{bmatrix} \boldsymbol{A}_{\text{pp}} \\ \boldsymbol{w}_{\text{pp2}}^T \end{bmatrix}, \quad \boldsymbol{b}_{\text{pp2}} = \begin{bmatrix} \boldsymbol{b}_{\text{pp}} \\ 0 \end{bmatrix}. \tag{42}$$

B. Line-Line Case

$\mathcal{F}_c^i = \mathcal{L}_c^i, \mathcal{F}_r^i = \mathcal{L}_r^i, \mathcal{F}_c^j = \mathcal{L}_c^j, \mathcal{F}_r^j = \mathcal{L}_r^j$.

In the case of two line pairs, given the consistent rotation $\boldsymbol{R} \in \mathcal{R}$, the consistent translation can be calculated by solving the following linear equations:

$$\begin{aligned} \left[\boldsymbol{v}_c^i\right]_\times\boldsymbol{t} &= \boldsymbol{R}\boldsymbol{u}_r^i - \boldsymbol{u}_c^i \\ \left[\boldsymbol{v}_c^j\right]_\times\boldsymbol{t} &= \boldsymbol{R}\boldsymbol{u}_r^j - \boldsymbol{u}_c^j. \end{aligned} \tag{43}$$

Rewrite (43) in a matrix form

$$\boldsymbol{A}_{\text{ll}}\boldsymbol{t} = \boldsymbol{b}_{\text{ll}} \tag{44}$$

$$\boldsymbol{A}_{\text{ll}} = \begin{bmatrix} \left[\boldsymbol{v}_c^i\right]_\times \\ \left[\boldsymbol{v}_c^j\right]_\times \end{bmatrix}, \quad \boldsymbol{b}_{\text{ll}} = \begin{bmatrix} \boldsymbol{R}\boldsymbol{u}_r^i - \boldsymbol{u}_c^i \\ \boldsymbol{R}\boldsymbol{u}_r^j - \boldsymbol{u}_c^j \end{bmatrix}. \tag{45}$$

If $\boldsymbol{v}_c^i = \boldsymbol{v}_c^j = \boldsymbol{v}_c$, i.e., the lines in the same coordinate system are parallel to each other, then $\operatorname{rank}(\boldsymbol{A}_{\text{ll}}) = 2$. The linear system (44) has solutions if and only if $\operatorname{rank}([\boldsymbol{A}_{\text{ll}}|\boldsymbol{b}_{\text{ll}}]) = 2$, which requires that

$$\boldsymbol{R}\boldsymbol{u}_r^i - \boldsymbol{u}_c^i = \boldsymbol{R}\boldsymbol{u}_r^j - \boldsymbol{u}_c^j. \tag{46}$$

According to the properties of the rotation matrix $\boldsymbol{R}$, (46) is equivalent to the following equations:

$$\left\|\boldsymbol{u}_r^i - \boldsymbol{u}_r^j\right\| = \left\|\boldsymbol{u}_c^i - \boldsymbol{u}_c^j\right\| \tag{47}$$

$$\boldsymbol{R} \cdot \frac{\boldsymbol{u}_r^i - \boldsymbol{u}_r^j}{\left\|\boldsymbol{u}_r^i - \boldsymbol{u}_r^j\right\|} = \frac{\boldsymbol{u}_c^i - \boldsymbol{u}_c^j}{\left\|\boldsymbol{u}_c^i - \boldsymbol{u}_c^j\right\|}. \tag{48}$$

As known from Algorithm 1, if $\boldsymbol{v}_c^i = \boldsymbol{v}_c^j$, the consistent rotation $\boldsymbol{R}$ has one DoF. Let $\boldsymbol{v}_r = \boldsymbol{v}_r^i = \boldsymbol{v}_r^j$. In this case, if (47) holds true, we need to solve for a rotation to satisfy (48) and $\boldsymbol{v}_c = \boldsymbol{R}\boldsymbol{v}_r$ simultaneously by

$$\textsc{RotationConsistency}\left(\boldsymbol{v}_c, \boldsymbol{v}_r, \frac{\boldsymbol{u}_c^i - \boldsymbol{u}_c^j}{\left\|\boldsymbol{u}_c^i - \boldsymbol{u}_c^j\right\|}, \frac{\boldsymbol{u}_r^i - \boldsymbol{u}_r^j}{\left\|\boldsymbol{u}_r^i - \boldsymbol{u}_r^j\right\|}\right).$$

If the returning value is not $\emptyset$, then $\mathcal{N}^i$ and $\mathcal{N}^j$ are translation consistent and the resultant translation has one DoF

$$\boldsymbol{t} = \boldsymbol{t}_{\text{ll1}} + \mu \boldsymbol{w}_{\text{ll1}} \tag{49}$$

where

$$\begin{aligned} \boldsymbol{w}_{\text{ll1}} &= \boldsymbol{v}_c, \ \mu \in \mathbb{R} \\ \boldsymbol{t}_{\text{ll1}} &= \left(\boldsymbol{A}_{\text{ll1}}^T \boldsymbol{A}_{\text{ll1}}\right)^{-1} \boldsymbol{A}_{\text{ll1}}^T \boldsymbol{b}_{\text{ll1}} \\ \boldsymbol{A}_{\text{ll1}} &= \begin{bmatrix} \boldsymbol{A}_{\text{ll}} \\ \boldsymbol{w}_{\text{ll1}}^T \end{bmatrix}, \quad \boldsymbol{b}_{\text{ll1}} = \begin{bmatrix} \boldsymbol{b}_{\text{ll}} \\ 0 \end{bmatrix}. \end{aligned} \tag{50}$$

If $\boldsymbol{v}_c^i \neq \boldsymbol{v}_c^j$, i.e., the lines in the same coordinate system are nonparallel, then $\text{rank}(\boldsymbol{A}_{\text{ll}}) = 3$. The linear system (44) has solutions if and only if $\text{rank}([\boldsymbol{A}_{\text{ll}}|\boldsymbol{b}_{\text{ll}}]) = 3$, which requires

$$\begin{aligned} l\left(\mathcal{L}_r^i, \mathcal{L}_r^j\right) - l\left(\mathcal{L}_c^i, \mathcal{L}_c^j\right) = \left(\boldsymbol{R}\boldsymbol{u}_r^i - \boldsymbol{u}_c^i\right)^T \boldsymbol{v}_c^j \\ + \left(\boldsymbol{R}\boldsymbol{u}_r^j - \boldsymbol{u}_c^j\right)^T \boldsymbol{v}_c^i = 0 \end{aligned} \tag{51}$$

where $l(\mathcal{L}^i, \mathcal{L}^j) = \boldsymbol{u}^{i^T} \boldsymbol{v}^j + \boldsymbol{u}^{j^T} \boldsymbol{v}^i$ represents the vertical distance between two 3-D lines. Criterion (51) means that the vertical distance between two lines in the current frame equals to that between the corresponding two lines in the reference frame. If (51) is satisfied, the translation is constrained and can be computed by

$$\boldsymbol{t} = \left(\boldsymbol{A}_{\text{ll}}^T \boldsymbol{A}_{\text{ll}}\right)^{-1} \boldsymbol{A}_{\text{ll}}^T \boldsymbol{b}_{\text{ll}}. \tag{52}$$

C. Plane-Line Case

$\mathcal{F}_c^i = \pi_c^i, \mathcal{F}_r^i = \pi_r^i, \mathcal{F}_c^j = \mathcal{L}_c^j, \mathcal{F}_r^j = \mathcal{L}_r^j$.

In this case, one pair of planes (π_c^i and π_r^i) and one pair of lines ($\mathcal{L}_c^j$ and $\mathcal{L}_r^j$) are considered. If their orientations are aligned by $\boldsymbol{R} \in \mathcal{R}$, the consistent translation that makes $\mathcal{N}^i$ and $\mathcal{N}^j$ internode consistent is computed by solving the following equation:

$$\begin{aligned} \boldsymbol{n}_c^{i^T} \boldsymbol{t} &= d_r^i - d_c^i \\ \left[\boldsymbol{v}_c^j\right]_\times \boldsymbol{t} &= \boldsymbol{R}\boldsymbol{u}_r^j - \boldsymbol{u}_c^j. \end{aligned} \tag{53}$$

Rewrite (53) in a matrix form as

$$\boldsymbol{A}_{\text{pl}} \boldsymbol{t} = \boldsymbol{b}_{\text{pl}} \tag{54}$$

$$\boldsymbol{A}_{\text{pl}} = \begin{bmatrix} \boldsymbol{n}_c^{i^T} \\ \left[\boldsymbol{v}_c^j\right]_\times \end{bmatrix}, \quad \boldsymbol{b}_{\text{pl}} = \begin{bmatrix} d_r^i - d_c^i \\ \boldsymbol{R}\boldsymbol{u}_r^j - \boldsymbol{u}_c^j \end{bmatrix}. \tag{55}$$

If $\boldsymbol{n}_c^{i^T} \boldsymbol{v}_c^j = 0$, i.e., the plane and the line in the same coordinate system are parallel to each other, then $\text{rank}(\boldsymbol{A}_{\text{pl}}) = 2$ because $\boldsymbol{n}_c^i$ is in the column space of $[\boldsymbol{v}_c^j]_\times$. Thus, $\text{rank}([\boldsymbol{A}_{\text{pl}}|\boldsymbol{b}_{\text{pl}}]) = 2$ holds true if

$$\begin{aligned} l\left(\pi_r^i, \mathcal{L}_r^j\right) - l\left(\pi_c^i, \mathcal{L}_c^j\right) = \boldsymbol{n}_c^{i^T} \left[\boldsymbol{v}_c^j\right]_\times \left(\boldsymbol{R}\boldsymbol{u}_r^j - \boldsymbol{u}_c^j\right) \\ + \left(d_r^i - d_c^i\right) = 0 \end{aligned} \tag{56}$$

where $l(\pi, \mathcal{L}) = \boldsymbol{n}^T [\boldsymbol{v}]_\times \boldsymbol{u} + d$ represents the vertical distance between π and $\mathcal{L}$ which are parallel to each other. In other words, if the vertical distance between the plane and the line in the current frame equals to that in the reference frame, the linear system (54) has solutions which have one DoF

$$\boldsymbol{t} = \boldsymbol{t}_{\text{pl1}} + \mu \boldsymbol{w}_{\text{pl1}} \tag{57}$$

where

$$\begin{aligned} \boldsymbol{w}_{\text{pl1}} &= \boldsymbol{v}_c^j, \ \mu \in \mathbb{R} \\ \boldsymbol{t}_{\text{pl1}} &= \left(\boldsymbol{A}_{\text{pl1}}^T \boldsymbol{A}_{\text{pl1}}\right)^{-1} \boldsymbol{A}_{\text{pl1}}^T \boldsymbol{b}_{\text{pl1}} \\ \boldsymbol{A}_{\text{pl1}} &= \begin{bmatrix} \boldsymbol{A}_{\text{pl}} \\ \boldsymbol{w}_{\text{pl1}}^T \end{bmatrix}, \quad \boldsymbol{b}_{\text{pl1}} = \begin{bmatrix} \boldsymbol{b}_{\text{pl}} \\ 0 \end{bmatrix}. \end{aligned} \tag{58}$$

If $\boldsymbol{n}_c^i = \boldsymbol{v}_c^j$, i.e., the line is vertical to the plane in the same coordinate system, then $\text{rank}(\boldsymbol{A}_{\text{pl}}) = 3$. As known from Algorithm 1, in this case, the consistent rotation has one DoF. Solving (54) directly yields

$$\begin{aligned} \boldsymbol{t} &= \left(d_r^i - d_c^i\right)\boldsymbol{n}_c^i - \left[\boldsymbol{v}_c^j\right]_\times \left(\boldsymbol{R}\boldsymbol{u}_r^j - \boldsymbol{u}_c^j\right) \\ &= \left(d_r^i - d_c^i\right)\boldsymbol{n}_c^i + \boldsymbol{v}_c^j \times \boldsymbol{u}_c^j + \boldsymbol{R}\left(\boldsymbol{u}_r^j \times \boldsymbol{v}_r^j\right) \\ &= \boldsymbol{t}_{\text{pl2}} + \boldsymbol{R}\boldsymbol{w}_{\text{pl2}}, \quad \boldsymbol{R} \in \mathcal{R}. \end{aligned} \tag{59}$$

It is obvious from (59) that the resultant translation $\boldsymbol{t}$ has one DoF φ as the rotation $\boldsymbol{R}$ does.

If $\boldsymbol{n}_c^{i^T} \boldsymbol{v}_c^j \neq 0$ and $\boldsymbol{n}_c^i \neq \boldsymbol{v}_c^j$, then $\text{rank}(\boldsymbol{A}) = 3$. In this case, $\text{rank}([\boldsymbol{A}|\boldsymbol{b}]) = 3$ holds true. Thus, the translation is constrained and can be computed directly by

$$\boldsymbol{t} = \left(\boldsymbol{A}_{\text{pl}}^T \boldsymbol{A}_{\text{pl}}\right)^{-1} \boldsymbol{A}_{\text{pl}}^T \boldsymbol{b}_{\text{pl}}. \tag{60}$$

References

[1] T. Ran, L. Yuan, J. Zhang, L. He, R. Huang, and J. Mei, "Not only look but infer: Multiple hypothesis clustering of data association inference for semantic SLAM," *IEEE Trans. Instrum. Meas.*, vol. 70, pp. 1–9, 2021.

[2] J. Wang, Z. Meng, and L. Wang, "Efficient probabilistic approach to range-only SLAM with a novel likelihood model," *IEEE Trans. Instrum. Meas.*, vol. 70, pp. 1–12, 2021.

[3] G. He, X. Yuan, Y. Zhuang, and H. Hu, "An integrated GNSS/LiDAR-SLAM pose estimation framework for large-scale map building in partially GNSS-denied environments," *IEEE Trans. Instrum. Meas.*, vol. 70, pp. 1–9, 2021.

[4] C. Campos, R. Elvira, J. J. G. Rodríguez, J. M. M. Montiel, and J. D. Tardós, "ORB-SLAM3: An accurate open-source library for visual, visual-inertial and multi-map SLAM," 2020, *arXiv:2007.11898*. [Online]. Available: http://arxiv.org/abs/2007.11898

[5] 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.

[6] R. Munguía and A. Grau, "Closing loops with a virtual sensor based on monocular SLAM," *IEEE Trans. Instrum. Meas.*, vol. 58, no. 8, pp. 2377–2384, Aug. 2009.

[7] F.-A. Moreno, D. Zuñiga-Noël, D. Scaramuzza, and J. Gonzalez-Jimenez, "PL-SLAM: A stereo SLAM system through the combination of points and line segments," *IEEE Trans. Robot.*, vol. 35, no. 3, pp. 734–746, Jun. 2019.

[8] P. F. Proença and Y. Gao, "Probabilistic combination of noisy points and planes for RGB-D odometry," in *Proc. Annu. Conf. Towards Auto. Robotic Syst.*, 2017, pp. 340–350.

[9] H. Gao, X. Zhang, J. Wen, J. Yuan, and Y. Fang, "Autonomous indoor exploration via polygon map construction and graph-based SLAM using directional endpoint features," *IEEE Trans. Autom. Sci. Eng.*, vol. 16, no. 4, pp. 1531–1542, Oct. 2019.

[10] B. Fang and Z. Zhan, "A visual SLAM method based on point-line fusion in weak-matching scene," *Int. J. Adv. Robotic Syst.*, vol. 17, no. 2, Mar. 2020, Art. no. 172988142090419.

[11] P. F. Proença and Y. Gao, "Probabilistic RGB-D odometry based on points, lines and planes under depth uncertainty," *Robot. Auto. Syst.*, vol. 104, pp. 25–39, Jun. 2018.

[12] C. Raposo, M. Antunes, and J. P. Barreto, "Piecewise-planar stereoscan: Sequential structure and motion using plane primitives," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 8, pp. 1918–1931, Aug. 2018.

[13] X. Zuo, X. Xie, Y. Liu, and G. Huang, "Robust visual SLAM with point and line features," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2017, pp. 1775–1782.

[14] Y. Taguchi, Y.-D. Jian, S. Ramalingam, and C. Feng, "Point-plane SLAM for hand-held 3D sensors," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2013, pp. 5182–5189.

[15] S. Yang and S. Scherer, "Monocular object and plane SLAM in structured environments," *IEEE Robot. Autom. Lett.*, vol. 4, no. 4, pp. 3145–3152, Oct. 2019.

[16] S. Yang, Y. Song, M. Kaess, and S. Scherer, "Pop-up SLAM: Semantic monocular plane SLAM for low-texture environments," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2016, pp. 1222–1229.

[17] S. Y. Bao, M. Bagra, Y.-W. Chao, and S. Savarese, "Semantic structure from motion with points, regions, and objects," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 2703–2710.

[18] F. Nardi, B. D. Corte, and G. Grisetti, "Unified representation and registration of heterogeneous sets of geometric primitives," *IEEE Robot. Autom. Lett.*, vol. 4, no. 2, pp. 625–632, Apr. 2019.

[19] A. Elqursh and A. Elgammal, "Line-based relative pose estimation," in *Proc. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2011, pp. 3049–3056.

[20] M. Hsiao, E. Westman, G. Zhang, and M. Kaess, "Keyframe-based dense planar SLAM," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2017, pp. 5110–5117.

[21] 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.

[22] M. A. Fischler and R. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[23] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. Cambridge, U.K.: Cambridge Univ. Press, 2004.

[24] W. Grimson, *Object Recognition by Computer: The Role of Geometric Constraints*. Cambridge, MA, USA: MIT Press, 1990.

[25] T. He and S. Hirose, "A global localization approach based on line-segment relation matching technique," *Robot. Auto. Syst.*, vol. 60, no. 1, pp. 95–112, Jan. 2012.

[26] X. Wang, J. Song, M. Feng, M. Chong, and J. Li, "Incremental mapping based on line-segments relation for mobile robot," in *Proc. 3rd Int. Conf. Robot. Autom. Eng. (ICRAE)*, 2018, pp. 65–70.

[27] K. O. Arras, J. A. Castellanos, M. Schilt, and R. Siegwart, "Feature-based multi-hypothesis localization and tracking using geometric constraints," *Robot. Auto. Syst.*, vol. 44, no. 1, pp. 41–53, Jul. 2003.

[28] J. Neira and J. D. Tardos, "Data association in stochastic mapping using the joint compatibility test," *IEEE Trans. Robot. Autom.*, vol. 17, no. 6, pp. 890–897, Dec. 2001.

[29] J. Neira, J. D. Tardos, and J. A. Castellanos, "Linear time vehicle relocation in SLAM," in *Proc. IEEE Int. Conf. Robot. Autom.*, vol. 1, Jun. 2003, pp. 427–433.

[30] Y. Li and E. B. Olson, "IPJC: The incremental posterior joint compatibility test for fast feature cloud matching," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Oct. 2012, pp. 3467–3474.

[31] X. Shen, E. Frazzoli, D. Rus, and M. H. Ang, "Fast joint compatibility branch and bound for feature cloud matching," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2016, pp. 1757–1764.

[32] J. Wang and B. Englot, "Robust exploration with multiple hypothesis data association," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2018, pp. 3537–3544.

[33] X. Li, Y. He, J. Lin, and X. Liu, "Leveraging planar regularities for point line visual-inertial odometry," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2021, pp. 5120–5127.

[34] X. Zhang, W. Wang, X. Qi, Z. Liao, and R. Wei, "Point-plane SLAM using supposed planes for indoor environments," *Sensors*, vol. 19, no. 17, p. 3795, Sep. 2019.

[35] Y. Yang, P. Geneva, X. Zuo, K. Eckenhoff, Y. Liu, and G. Huang, "Tightly-coupled aided inertial navigation with point and plane features," in *Proc. Int. Conf. Robot. Autom. (ICRA)*, May 2019, pp. 6094–6100.

[36] Y. Yang and G. Huang, "Observability analysis of aided INS with heterogeneous features of points, lines, and planes," *IEEE Trans. Robot.*, vol. 35, no. 6, pp. 1399–1418, Dec. 2019.

[37] M. Kaess, "Simultaneous localization and mapping with infinite planes," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2015, pp. 4605–4611.

[38] F. Zheng, G. Tsai, Z. Zhang, S. Liu, C.-C. Chu, and H. Hu, "Trifo-VIO: Robust and efficient stereo visual inertial odometry using points and lines," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2018, pp. 3686–3693.

[39] Q. Fu *et al.*, "PL-VINS: Real-time monocular visual-inertial SLAM with point and line features," 2020, *arXiv:2009.07462v1*. [Online]. Available: https://arxiv.org/abs/2009.07462v1

[40] H. Li, J. Yao, X. Lu, and J. Wu, "Combining points and lines for camera pose estimation and optimization in monocular visual odometry," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2017, pp. 1289–1296.

[41] 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.

[42] A. Bartoli and P. Sturm, "Structure-from-motion using lines: Representation, triangulation, and bundle adjustment," *Comput. Vis. Image Understand.*, vol. 100, no. 3, pp. 416–441, 2005.

[43] A. J. B. Trevor, S. Gedikli, R. B. Rusu, and H. I. Christensen, "Efficient organized point cloud segmentation with connected components," in *Proc. Semantic Perception Mapping Explor. (SPME)*, 2013, pp. 1–6.

[44] 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.

[45] P. C. Gaston and T. Lozano-Prez, "Tactile recognition and localization using object models: The case of polyhedra on a plane," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-6, pp. 257–265, 1983.

[46] F. S. Grassia, "Practical parameterization of rotations using the exponential map," *J. Graph. Tools*, vol. 3, no. 3, pp. 29–48, 1998.

[47] 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. Robot Syst. (IROS)*, Vilamoura, Portugal, Oct. 2012, pp. 573–580.

[48] A. Handa, T. Whelan, J. McDonald, and A. J. Davison, "A benchmark for RGB-D visual odometry, 3D reconstruction and slam," in *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, May 2014, pp. 1524–1531.

[49] Q. Sun, J. Yuan, X. Zhang, and F. Sun, "RGB-D SLAM in indoor environments with STING-based plane feature extraction," *IEEE/ASME Trans. Mechatronics*, vol. 23, no. 3, pp. 1071–1082, Jun. 2018.

[50] Y. Zhou, H. Li, and L. Kneip, "Canny-VO: Visual odometry with RGB-D cameras based on geometric 3D-2D edge alignment," *IEEE Trans. Robot.*, vol. 35, pp. 184–199, Feb. 2019.

[51] 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.

[52] L. Ma, C. Kerl, J. Stückler, and D. Cremers, "CPA-SLAM: Consistent plane-model alignment for direct RGB-D SLAM," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2016, pp. 1285–1291.

[53] 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.

[54] E. Ataer-Cansizoglu, Y. Taguchi, and S. Ramalingam, "Pinpoint SLAM: A hybrid of 2D and 3D simultaneous localization and mapping for RGB-D sensors," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2016, pp. 1300–1307.

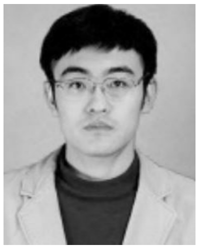
[55] T. Whelan, R. F. Salas-Moreno, B. Glocker, A. J. Davison, and S. Leutenegger, "ElasticFusion: Real-time dense SLAM and light source estimation," *Int. J. Robot. Res.*, vol. 35, no. 14, pp. 1697–1716, 2016.

[56] L. Han, L. Xu, D. Bobkov, E. Steinbach, and L. Fang, "Real-time global registration for globally consistent RGB-D SLAM," *IEEE Trans. Robot.*, vol. 35, no. 2, pp. 498–508, Apr. 2019.

[57] Z. Yan, M. Ye, and L. Ren, "Dense visual SLAM with probabilistic surfel map," *IEEE Trans. Vis. Comput. Graphics*, vol. 23, no. 11, pp. 2389–2398, Nov. 2017.

Qinxuan Sun received the B.Sc. degree in electronic information engineering from Beijing University of Aeronautics and Astronautics, Beijing, China, in 2013, and the M.Sc. degree in control theory and control engineering from Nankai University, Tianjin, China, in 2016, where she is currently pursuing the Ph.D. degree.

Her current research interests include mobile robot navigation and SLAM.



Jing Yuan (Member, IEEE) received the B.Sc. degree in automatic control and the Ph.D. degree in control theory and control engineering from Nankai University, Tianjin, China, in 2002 and 2007, respectively.

He has been with the Department of Automation, Nankai University, since 2007, where he is currently a Professor. His current research interests include robotic control, target tracking, and SLAM.

Xuebo Zhang (Senior Member, IEEE) received the B.Eng. degree in automation from Tianjin University, Tianjin, China, in 2006, and the Ph.D. degree in control theory and control engineering from Nankai University, Tianjin, in 2011.

He is currently a Professor with the Institute of Robotics and Automatic Information System, Nankai University. His current research interests include motion planning, visual servoing, and SLAM.

Dr. Zhang is a Technical Editor of the IEEE/ASME TRANSACTIONS ON MECHATRONICS and an Associate Editor of the *ASME Journal of Dynamic Systems, Measurement, and Control.*