IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

PAPL-SLAM: Principal Axis-Anchored Monocular Point-Line SLAM

Guanghao Li[1], Yu Cao[2], Qi Chen[2], Xin Gao[1], Yifan Yang[1], Jian Pu[1,*]

***Abstract*—In point-line Simultaneous Localization and Mapping (SLAM) systems, the utilization of line structural information and the optimization of lines are two significant problems. The former is usually addressed through structural regularities, while the latter typically involves using minimal parameter representations of lines in optimization. However, separating these two steps leads to the loss of constraint information to each other. To solve both problems, we anchor lines with similar directions to one principal axis. Precisely, our method models the line-axis probabilistic data association using the Expectation Maximization (EM) algorithm and provides the pipelines for axis creation, updating, and optimization, enhancing the system's robustness and avoiding mismatch. Our system can optimize n co-directional lines with only $n+2$ parameters, significantly reducing the number of line parameters to be optimized and enabling rapid mapping and tracking. Additionally, considering that most real-world scenes conform to the Atlanta World (AW) hypothesis, we provide an AW constraint by detecting structural lines based on vertical priors and vanishing points. Experimental results and ablation studies on various indoor and outdoor datasets demonstrate the effectiveness of our system.**

***Index Terms*—Line Features, Data Association, Bundle Adjustment (BA), Expectation Maximization (EM), Simultaneous Localization and Mapping (SLAM).**

I. Introduction

SIMULTANEOUS Localization and Mapping (SLAM), as a classical problem, has seen significant development over the past two decades. Among the numerous SLAM systems [1]–[3], monocular Vision SLAM (VSLAM) stands out as a classic system, widely adopted for its portability and low cost [4]. However, most VSLAM systems [5]–[8] depend on point features, which perform poorly in varying lighting conditions and low-texture areas (typically artificial environments). Consequently, recent years have seen a surge in research on point-line VSLAM systems [9], [10], which are useful [11] for the complex environment. In these systems, the additional degree of freedom in line features compared to point features provides extra structural information but also introduces challenges in optimization. Therefore, utilizing the structural information provided by line features and efficiently optimizing lines are two significant problems that are usually seen as separate.

The structural information of lines is evident as many line features exhibit the same (or similar) orientation, thus giving rise to SLAM systems based on hypotheses like the Manhattan World [12], Mixture of Manhattan World [13], Atlanta World [14], and Hong Kong World assumptions [15]. These systems leverage specific patterns to impose constraints on algorithm design, yielding effective results. However, these systems do not integrate the structural information provided by world assumptions with line optimization and, therefore, do not fully utilize the structural information to constrain tracking accuracy and global consistency.

The optimization of a line depends on its representation. Different representations exhibit different characteristics in VSLAM. The representation of a line has two categories: over-parameterized and minimal parameter representations. Over-parameterized representations, such as two endpoints, Edgelets [16], and Plücker coordinates [17], [18], are intuitive but require additional constraints during Bundle Adjustment (BA). Furthermore, over-parameterized representation introduces extra computational overhead, numerical instability, and even unsuitability for nonlinear optimization. On the other hand, minimal parameter representations like Orthogonal Representation [19], Cayley Representation [20], and the No Singularities and Special Cases [21] utilize four parameters to describe a line, which is less intuitive compared with over-parameterized representation. They do not require additional constraints during optimization, but their capability for optimizing line orientation is limited [22]. Besides, regardless of the representation used, it needs at least four parameters for a line and much time to reach a relatively good accuracy.

Integrating structural information and optimization in a unified framework is a natural approach, as optimization can leverage global structural regularities [23], while structural constraints can further reduce the number of parameters in optimization [24]. However, direct incorporation of structural constraints into optimization remains challenging due to inherent incompatibilities [22]. A promising strategy to bridge this gap is to adopt an appropriate line representation that naturally encodes structural priors while maintaining optimization efficiency, which serves as the foundation of our work.

We present PAPL-SLAM, a monocular VSLAM point-line system that uses a principal axis-anchored line representation to incorporate structural information into optimization. Specifically, we restore the structural line features by utilizing the orientation of its anchored principal axis and the inverse depth of the midpoint of the observed line segment

Manuscript received: March, 25, 2025; Revised June, 19, 2025; Accepted July, 11, 2025.

This paper was recommended for publication by Editor Pascal Vasseur upon evaluation of the Associate Editor and Reviewers' comments. This work was partially supported by the Computing for the Future at Fudan (CFFF). (Guanghao Li, Yu Cao, and Qi Chen contributed equally to this work.) (*Corresponding author: Jian Pu)

[1]Guanghao Li, Xin Gao, Yifan Yang, and Jian Pu are with the Institute of Science and Technology for Brain-Inspired Intelligence (ISTBI), Fudan University, Shanghai, 200433, China (E-mails: {ghli22, gaoxin23, yifyang23}@m.fudan.edu.cn, jianpu@fudan.edu.cn).

[2]Qi Chen and Yu Cao are with Shanghai Key Lab of Intelligent Information Processing and School of Computer Science, Fudan University, Shanghai 200433, China (E-mail: {qichen21, caoyu21}@m.fudan.edu.cn).

Digital Object Identifier (DOI): see top of this page.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

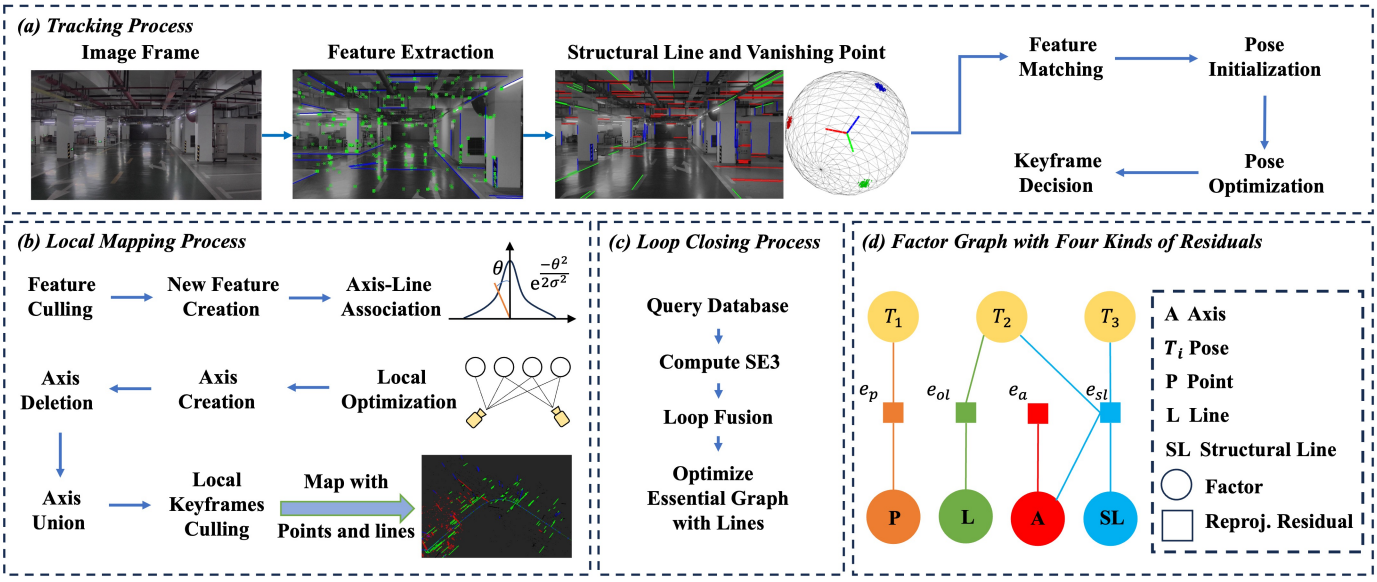

Fig. 1. The architecture of our system. Our system consists of tracking, local mapping, and loop closure threads. (a) shows the tracking thread workflow, where we detect structural lines in the scene and perform pose optimization. (b) shows the local mapping thread, where we perform principal axis management and local optimization. (c) show the loop closing process, using structural lines to optimize the essential graph. (d) shows our factor graph for BA, using a three-parameter optimization method based on principal axes and traditional point and line feature optimization.

in the reference keyframe, which allows us to optimize n co-directional lines using only $n+2$ parameters. To handle the mismatch problem, we use expectation maximization to model the line-axis probabilistic data association. We also design pipelines for axis creation, updating, and optimization to form a complete line-axis optimization algorithm. Besides, to prove that our line representation enables easy extension to different world hypotheses, we add the Atlanta World hypothesis to our VSLAM system, which achieves better accuracy with this additional constraint. The main contributions of this paper are summarized as follows:

- We present a monocular point-line VSLAM system with well-designed modules. Extensive experiments on our system demonstrate its robustness and precision.
- We incorporate the scene's structural information into the optimization process using our principal axis-anchored line representation, enhancing the tracking accuracy and presenting a more readable map representation.
- We provide a complete line-axis management pipeline that includes creating, updating, probabilistically associating, and optimizing principal axes, enhancing the system's robustness while maintaining high running speed.
- We make an additional constraint for scenes commonly encountered under the Atlanta World hypothesis, improving the tracking accuracy of artificial environments.

II. Related Works

Here, we briefly introduce representative SLAM systems. For a more detailed review, please refer to SLAM surveys [25], [26].

A. Point SLAM System

Most systems adopted point features due to their simplicity and practicality. There were many methods to detect the characteristics of the points, such as the Harris corner, Shi-Tomasi corner, SIFT, FAST, SURF, Brief, and ORB. MonoSLAM [4], as the first real-time monocular VSLAM algorithm, used Shi-Tomasi corners for tracking in the frontend and employed an Extended Kalman Filter (EKF) for optimization in the backend. PTAM [27] divided VSLAM into two threads: mapping and tracking. The tracking thread used FAST corners for pose estimation, while the mapping thread replaced the EKF with a nonlinear optimization algorithm. LSD-SLAM [28] used a direct method by optimizing pixel intensities and performed loop closure with feature points. The ORB series [5], [6], [8] adopted PTAM's dual-thread approach and introduced a loop closure detection thread. It used ORB features [29] for tracking and DBoW [30] for loop closure detection, making it one of the most influential SLAM systems today. Edge SLAM [31] detected edge points from images and tracked those using optical flow. However, point features cannot be easily recognized and matched in areas with low textures and lighting variations. In such regions, fully utilizing the scene's structural information can help improve localization and mapping.

B. Point-Line SLAM System

Line features provide more structural information to adapt to complex environments than point features. Some VSLAM systems [32]–[37] used line features to add additional constraints to the scene, while others [24], [38], [39] went further by using scene assumptions to enhance these constraints. For the former, [40] used an EKF to jointly optimize point features and line features represented by two endpoints. PL-SLAM [9] represented a 3D line with its two 3D endpoints and introduced a new reprojection error. EDPLVO [41] used the inverse depth of two endpoints for optimization. However, these systems did not utilize the directional information of the lines effectively.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

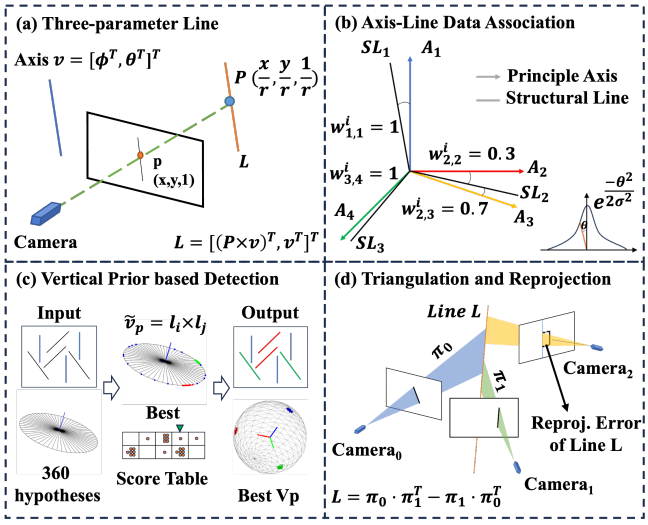

Fig. 2. Principal axis and line handling algorithms. (a) illustrates the definition of our three-parameter line representation. (b) defines our principal axis probabilistic association model. (c) shows the structural line and vanishing point detection strategy based on the vertical prior. (d) demonstrates the triangulation process of 3D lines and the reprojection error during optimization.

For the latter, many systems assumed the Manhattan World hypothesis [42]–[47], often optimizing the rotation matrix first and then the translation vector, but most were suited for indoor environments. Some systems were also based on the Atlanta World [11], [23], [48] or Hong Kong World [15] hypotheses, providing a more general representation of structured scenes. UV-SLAM [22] used structural regularities (vanishing point factor) without any constraints, making it more broadly applicable. However, the systems above do not effectively represent a 3D line with directional information in BA. Our system achieves this representation while also providing fast and accurate pose estimation.

III. Method

Given the known camera intrinsic matrix $\boldsymbol{K} \in \boldsymbol{R}^{3\times3}$, the input to our system is a sequence of image frames $\{\boldsymbol{I}_t\}_{t=1}^{N_I}$, and the output consists of estimated poses $\boldsymbol{T}_{cw}$ and a map with point features $\boldsymbol{P}_n$ and line features $\boldsymbol{L}_n$ in three dimension space. The pose $\boldsymbol{T}_{cw}$ is an element of $SE(3)$ (the special Euclidean group), comprising a rotational component $\boldsymbol{R}_{cw} \in SO(3)$ (the special orthogonal group) and a translational component $\boldsymbol{t}_{cw} \in R^3$. We propose PAPL-SLAM to solve this state estimation problem.

Fig. 1 illustrates the framework of PAPL-SLAM. We develop our system based on VPL-SLAM [23] and ORB-SLAM2 [6], which consist of three main components: tracking, local mapping, and loop closing. The tracking thread estimates the relative pose between the frame and its recent keyframe. Local mapping initializes new features and performs global BA between keyframes, while loop closing tries to detect a loop in the feature map and perform global loop BA. Feature is an essential element that transfers between these components, which plays a vital role in the system pipeline. Unlike ORB-SLAM2 [6], which creates and optimizes point features in the three components above, we add structural lines with multiple directions as a new feature. Therefore, we will first introduce the line representation (Sec. III-A) and line-axis management pipeline (Sec. III-B). To prove that our system can easily extend to different world hypotheses, we add one more constraint under the Atlanta World hypothesis in Sec. III-C. Finally, Sec. III-D introduces the objective function in the BA optimization of our whole system.

A. Point-Line System Model

In our system, we focus on the representation of lines and take the representation of points from ORB-SLAM2 [6]. We use a three-parameter representation to depict structural lines and an orthogonal representation for temporary non-structural lines during optimization. For their intuitive representation, we use end-point representation and Plücker coordinates in the non-optimization parts.

1) Over-parameterized Representation: Two distinct points $\boldsymbol{P}_1, \boldsymbol{P}_2$ in three-dimensional space can determine a 3D line $\boldsymbol{L}$, which is useful for visualization. We can also represent a 3D line $\boldsymbol{L}$ by its plücker representation, which is defined by two points on the line or by the direction vector $\boldsymbol{v}$ and one point $\boldsymbol{P}_3$ on the line:

$$\boldsymbol{L} = \begin{bmatrix} \boldsymbol{n} \\ \boldsymbol{v} \end{bmatrix} = \begin{bmatrix} \boldsymbol{P}_1 \times \boldsymbol{P}_2 \\ \boldsymbol{P}_2 - \boldsymbol{P}_1 \end{bmatrix} = \begin{bmatrix} \boldsymbol{P}_3 \times \boldsymbol{v} \\ \boldsymbol{v} \end{bmatrix}, \tag{1}$$

where $\boldsymbol{n} \in R^3$ is the normal vector of the plane determined by the line and the origin.

2) Orthogonal Representation: We adopt a three-dimensional rotation matrix $\boldsymbol{U} \in SO(3)$ (Eq. 2) to represent the transformation from the original coordinate system to a local coordinate system defined by the Plücker representation of a line. This local frame is constructed from the normalized normal vector $\boldsymbol{n}$, the normalized direction vector $\boldsymbol{v}$, and their normalized cross product. This matrix $\boldsymbol{U}$ captures the orientation of the line:

$$\boldsymbol{U} = \left[\frac{\boldsymbol{n}}{\|\boldsymbol{n}\|}, \frac{\boldsymbol{v}}{\|\boldsymbol{v}\|}, \frac{\boldsymbol{n} \times \boldsymbol{v}}{\|\boldsymbol{n} \times \boldsymbol{v}\|} \right]. \tag{2}$$

To represent the magnitude ratio between the normal vector and the direction vector—which determines the line's distance from the origin—we introduce a second rotation matrix $\boldsymbol{W} \in SO(2)$ (Eq. 3). This matrix is constructed using the norms of $\boldsymbol{n}$ and $\boldsymbol{v}$:

$$\boldsymbol{W} = \boldsymbol{R}(\phi) = \frac{1}{\sqrt{\|\boldsymbol{n}\|^2 + \|\boldsymbol{v}\|^2}} \begin{bmatrix} \|\boldsymbol{n}\| & -\|\boldsymbol{v}\| \\ \|\boldsymbol{v}\| & \|\boldsymbol{n}\| \end{bmatrix}. \tag{3}$$

Together, $\boldsymbol{U}$ and $\boldsymbol{W}$ form the orthogonal representation [19] of the line, denoted as $(\boldsymbol{U}, \boldsymbol{W}) = (\boldsymbol{R}(\boldsymbol{\psi}_o), \boldsymbol{R}(\phi_o)) \in SO(3) \times SO(2)$, where $\boldsymbol{\psi}_o \in \mathbb{R}^3$ is a 3D rotation vector representing the line's orientation, and $\phi_o \in \mathbb{R}$ is a 2D rotation angle reflecting the line's distance from the origin. This representation has a total of four degrees of freedom.

The orthogonal representation can be converted back to the Plücker representation via:

$$\boldsymbol{L} = \begin{bmatrix} \cos(\phi_o)\boldsymbol{U}^T_{:,1} \\ \sin(\phi_o)\boldsymbol{U}^T_{:,2} \end{bmatrix}, \tag{4}$$

where $\boldsymbol{U}_{:,i}$ denotes the i-th column of the matrix $\boldsymbol{U}$.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

3) Three-parameter Representation: If a line associates with an axis, it is called a structural line and can be represented by three parameters. For structural line, $\boldsymbol{L}$, the inverse depth r of the midpoint of the line segment in its reference keyframe and the direction $\boldsymbol{v}$ of its anchored principal axis can determine its Plücker representation (Fig. 2 (a)):

$$\boldsymbol{L} = \begin{bmatrix} \boldsymbol{P}^w \times \boldsymbol{v} \\ \boldsymbol{v} \end{bmatrix}, \quad \tilde{\boldsymbol{P}}^w = \boldsymbol{T}_{wc}\tilde{\boldsymbol{P}}^c, \quad \boldsymbol{P}^c = \boldsymbol{K}^{-1}\tilde{\boldsymbol{p}}/r. \quad (5)$$

$\boldsymbol{P}^w$ and $\boldsymbol{P}^c$ represent the midpoint in the world coordinate system and the camera coordinate system, respectively, while $\boldsymbol{p}$ represents the coordinates in the pixel coordinate system. The tilde symbol $\tilde{(\cdot)}$ above a letter indicates the corresponding homogeneous coordinates. Additionally, in order to avoid additional constraints during optimization, we use the latitude and longitude (ϕ_p, θ_p) to represent the direction vector of the principal axis $\boldsymbol{v} = [v_x, v_y, v_z]^T$:

$$\begin{cases} \phi_p = \arccos\left(v_z / \sqrt{v_x^2 + v_y^2 + v_z^2}\right) \\ \theta_p = \arctan\left(v_x, v_y\right) + \pi \end{cases}. \quad (6)$$

B. Line-Axis Management

Designing a management pipeline for the line-axis representation is necessary to provide a more robust system and handle the mismatch problem. In the following, we will introduce the creation, probabilistically associating and updating of lines and axes.

1) Creation: When the number of unclassified structural lines in the scene reaches a certain threshold τ_n, we identify potential principal axes among them. We model the process of finding principal axes to maximize a density function of direction. Specifically, we use the mean shift [49] algorithm to cluster the direction vectors $\boldsymbol{v}$ of all unclassified lines and identify the centers $m(\boldsymbol{v})$ among them:

$$\begin{aligned} m(\boldsymbol{v}) &= \frac{\sum_{\boldsymbol{v_i} \in \mathcal{D}_v} \mathcal{F}(\angle(\boldsymbol{v_i}, \boldsymbol{v}))\boldsymbol{v}_i}{\sum_{\boldsymbol{v_i} \in \mathcal{D}_v} \mathcal{F}(\angle(\boldsymbol{v_i}, \boldsymbol{v}))}, \\ \mathcal{F}\left(\angle(\boldsymbol{v_i}, \boldsymbol{v})\right) &= e^{-c\|\angle(\boldsymbol{v_i}, \boldsymbol{v})\|^2}, \end{aligned} \quad (7)$$

where $\mathcal{D}_v$ represents the set of other direction vectors within a certain angle τ_v of direction vector $\boldsymbol{v}$, c is the parameter to control the influence range of the kernel function, and $\angle(\boldsymbol{v_i}, \boldsymbol{v})$ is the function used to calculate the angle between two vectors. $\mathcal{F}(\cdot)$ is the Gaussian kernel function used to compute the weights based on different angles. We iteratively optimize Eq. 7 until the angle difference between the updated line and the previous line is smaller than a threshold τ_m, or the number of iterations exceeds 10.

For potential new principal axes, we first compare them with existing axes. The new axis is discarded if the angle difference is smaller than 10°. We then calculate the average angle difference between the related unclassified structural lines of the potential axis and the potential axis to identify the optimal principal axis. Additionally, we calculate the ratio of the number of non-structural lines to the number of structural lines in the current local map scene, and only if this ratio exceeds 0.6 will a new principal axis be created.

2) Data Association: The increase in principal axes and the drift in estimated poses can lead to incorrect line–principal axis association. To mitigate the impact of such incorrect associations and inspired by the approach to data association in semantic SLAM [50], we develop a probabilistic model for the principal axis–structural line association.

Given the estimated pose set $\mathcal{T} \triangleq \{\boldsymbol{T}_{wc,t}\}_{t=1}^{N_I}$, principal axis set $\mathcal{A} \triangleq \{\boldsymbol{A}_i\}_{m=1}^{N_A}$, and the set of structural lines $\mathcal{L} \triangleq \{\boldsymbol{L}_k\}_{k=1}^{N_L}$ to be associated, the data association $\mathcal{D} \triangleq \{\alpha_k\}_{k=1}^{N_L}$, which indicates the association of structural line $\boldsymbol{L}_k$ with principal axis $\boldsymbol{A}_{\alpha_k}$, is estimated. Specifically, with states $\mathcal{T}^i$ and $\mathcal{A}^i$ on step i, unlike previous hard associations, we calculate an optimal estimate $\mathcal{T}^{i+1}$, $\mathcal{A}^{i+1}$ by maximizing the expected measurement likelihood using EM algorithm, which considers the density distribution of all possible $\mathcal{D}$:

$$\begin{aligned} &\arg\max_{\mathcal{T},\mathcal{A}} \mathbb{E}_{\mathcal{D}}\left[\log p(\mathcal{L} \mid \mathcal{T}, \mathcal{A}, \mathcal{D}) \mid \mathcal{T}^i, \mathcal{A}^i, \mathcal{L}\right] \\ &= \arg\max_{\mathcal{T},\mathcal{A}} \sum_{\mathcal{D}\in\mathbb{D}} p_v\left(\mathcal{D} \mid \mathcal{T}^i, \mathcal{A}^i, \mathcal{L}\right) \log p(\mathcal{L} \mid \mathcal{T}, \mathcal{A}, \mathcal{D}) \\ &= \arg\max_{\mathcal{T},\mathcal{A}} \sum_{\mathcal{D}\in\mathbb{D}} \sum_{k=1}^{N_L} p_v\left(\mathcal{D} \mid \mathcal{T}^i, \mathcal{A}^i, \mathcal{L}\right) \log p\left(\boldsymbol{L}_k \mid \boldsymbol{\mathcal{T}}, \boldsymbol{A}_{\alpha_k}\right) \\ &= \arg\max_{\mathcal{T},\mathcal{A}} \sum_{k=1}^{N_L} \sum_{j=1}^{N_A} w_{kj}^i \log p\left(L_k \mid \mathcal{T}, A_j\right), \\ &w_{kj}^i = \sum_{\mathcal{D}\in\mathbb{D}(k,j)} \frac{p_v\left(\mathcal{L} \mid \mathcal{T}^i, \mathcal{A}^i, \mathcal{D}\right)}{\sum_{\mathcal{D}\in\mathbb{D}} p_v\left(\mathcal{L} \mid \mathcal{T}^i, \mathcal{A}^i, \mathcal{D}\right)}, \end{aligned} \quad (8)$$

where $\mathbb{D}$ is the space of all possible values of $\mathcal{D}$, and w_{kj}^i represents the combined possibility of the set $\mathbb{D}(k, j)$, which contains all possible associations where structural line $\boldsymbol{L}_k$ is associated with principal axis $\boldsymbol{A}_j$.

Fig. 2 (b) shows the data association process. To reduce computational load, we first calculate the angle between the direction vector of each structural line $\boldsymbol{L}_k$ and the direction of the principal axis $\boldsymbol{A}_m$. A relatively loose threshold filters out lines with large angles, whose weight $w_{k,m}$ are 0. We assume the probability $p_v\left(\boldsymbol{L}_k \mid \boldsymbol{\mathcal{T}}, \boldsymbol{A}_j\right)$ that a line belongs to an axis follows a normal distribution with a mean of 0°. The input to $p_v\left(\boldsymbol{L}_k \mid \boldsymbol{\mathcal{T}}, \boldsymbol{A}_j\right)$ is the angle difference between the average vanishing point direction across all frames where the structural line $\boldsymbol{L}_k$ is observed and the principal axis. Besides, $p_v\left(\mathcal{L} \mid \mathcal{T}^i, \mathcal{A}^i, \mathcal{D}\right)$ is different from $p\left(\mathcal{L} \mid \mathcal{T}, \mathcal{A}, \mathcal{D}\right)$ because the former only considers the directional information for line-axis association. In contrast, the latter considers projection, which includes direction and position, to optimize the pose and direction of the axis properly.

3) Update: We calculate the angle difference between the principal axis $\boldsymbol{A}_m$ before and after the BA. If the angle difference exceeds τ_{diff}, the direction of the principal axis after the BA is adopted.

C. Atlanta Wold (AW) Constraint

We use EDLines [51] to detect line segments, as it can detect more structural lines [23] with lower noise compared to LSD [52]. After detecting the line segments, we follow

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

the approach outlined in [23] for line segment fusion and matching. Finally, we extract vanishing points and structural lines from the images similar to the method described in [53]. Given that most structured scenes in everyday life conform to the AW hypothesis, we additionally provide a structural line and vanishing point detection algorithm based on vertical priors (Fig. 2 (c)).

We modify the vanishing point detection algorithm proposed in [53]. In the early stages of system operation, we extract vertical lines from the images at the frontend following the method outlined in [23]. The system begins extracting horizontal structural lines once the vertical principal axis $\boldsymbol{v}_a$ is initialized.

To improve robustness, we extract two dominant horizontal vanishing point directions in each frame based the known vertical principal axis $\boldsymbol{v}_a$. We first generate all the possible proposals for the two horizontal directions and then select the best proposal. The two directions lie on the plane $\boldsymbol{\pi}_a$ perpendicular to $\boldsymbol{v}_a$. Assuming $\boldsymbol{v}_a = [\sin a \sin b, \sin a \cos b, \cos a]^\top$ (a, b are angles that represents the direction in 3D space), we can derive a direction vector $\boldsymbol{v}_h = [\cos b, -\sin b, 0]^\top$ perpendicular to $\boldsymbol{v}_a$. The third direction can be calculated by their cross product. In this way, we obtain a proposal $\boldsymbol{h}_i = \{\boldsymbol{v}_a, \boldsymbol{v}_h, \boldsymbol{v}_a \times \boldsymbol{v}_h\}$. To get all the proposals, we can rotate $\boldsymbol{v}_h$ around $\boldsymbol{v}_a$ at different angles θ_i with step 1°, which generate $\boldsymbol{v}_{h,\theta_i}$ and all proposals $\boldsymbol{h}_i = \{\boldsymbol{v}_a, \boldsymbol{v}_{h,\theta_i}, \boldsymbol{v}_a \times \boldsymbol{v}_{h,\theta_i}\}_{i=1}^{360}$. Subsequently, the proposals $\{\boldsymbol{h}_i\}_{i=1}^{N}$ are scored, and the optimal proposal is selected. Through the vertical axis prior, the number of proposals is significantly reduced from the original 37,800 proposals [53] down to 360 proposals.

Further refinement is necessary because the number of samples limits the accuracy of the initial proposal. We use the distance from the lines to the two optimal vanishing points to find the corresponding line sets $\boldsymbol{\mathcal{C}}_0, \boldsymbol{\mathcal{C}}_1$. For each set $\boldsymbol{\mathcal{C}}$, we construct the matrix $\boldsymbol{M}$, where $\boldsymbol{M}_i = \tilde{\boldsymbol{s}}_i \times \tilde{\boldsymbol{e}}_i$ represents the parametric equation of the line segment in the row i, with $\tilde{\boldsymbol{s}}_i$ and $\tilde{\boldsymbol{e}}_i$ being the homogeneous coordinates of the observed line segment endpoints. The solution to the matrix equation $\boldsymbol{M}\boldsymbol{x} = 0$ represents the intersection point of the line segments in the set. By performing Singular Value Decomposition (SVD) on the matrix $\boldsymbol{M}$, we obtain the final vanishing point vp_{fine}. After optimization, the refined vanishing point reclassifies the structural line segments.

D. Optimization

We focus on optimizing lines in the map, while optimization of points can be referenced from ORB-SLAM2 [6]. For lines that are not yet associated with a principal axis, we initially treat their vanishing point direction in the corresponding keyframe as a temporary principal axis. In their first optimization, we optimize the lines using our three-parameter representation. After their first optimization, we switch to using the orthogonal representation to conduct the subsequent optimizations. This is done until the line is associated with at least one true principal axis. Once the line has been successfully associated with a principal axis, we revert back to the three-parameter optimization in the final stage. This three-stage optimization approach is particularly beneficial in the first stage. It helps mitigate the challenges associated with optimizing line directions using the orthogonal representation [22]. Furthermore, the approach lays a solid foundation for ensuring correct associations in later stages, making the overall optimization process more robust and accurate.

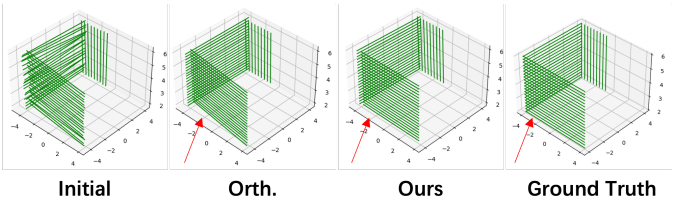

Fig. 3. Qualitative comparison between our representation and the orthogonal representation before and after optimization on the toy dataset, under given pose and observation perturbations.

TABLE I
RESULTS ON SYNTHETIC DATA

Metric	Fix Pose			Small Pose Noise			Large Pose Noise		
	2-p	4-p	3-p	2-p	4-p	3-p	2-p	4-p	3-p
Time (S)	0.956	1.097	**0.251**	5.977	7.359	**4.686**	6.934	8.767	**5.100**
Error$_l$ (CM)	0.014	**0.006**	0.009	0.596	0.942	**0.314**	6.577	5.144	**3.643**
Trans. (CM)	**0**	**0**	**0**	0.107	0.109	**0.072**	0.787	0.775	**0.747**
Rot. (Degree)	**0**	**0**	**0**	0.251	0.194	**0.173**	1.093	1.125	**0.885**

Error$_l$ represents the residual norm in the Plücker representation. "Trans." and "Rot." refer to the ATE RMSE metrics for the translation and rotation components of the pose, respectively.

1) Reprojection Error: Fig.2 (d) shows the process of 3D line triangulation and its reprojection error in the image. For a 3D line $\boldsymbol{L_w}$, we first translate it from the world coordinate to the current frame coordinate:

$$\boldsymbol{L_c} = \boldsymbol{T}_{cw}\boldsymbol{L}_w = \begin{bmatrix} \boldsymbol{R}_{cw} & [\boldsymbol{t}_{cw}]_\times \boldsymbol{R}_{cw} \\ \boldsymbol{0} & \boldsymbol{R}_{cw} \end{bmatrix} \boldsymbol{L_w} = \begin{bmatrix} \boldsymbol{n}_c \\ \boldsymbol{v}_c \end{bmatrix}, \quad (9)$$

where $[\cdot]_\times$ is the symbol for the antisymmetric matrix transformation for a vector. Then, we project the line $\boldsymbol{L}_c$ onto the pixel plane and calculate the reprojection error as the distance from the endpoints $\boldsymbol{p}_1, \boldsymbol{p}_2$ of the line segment to the projected line:

$$\boldsymbol{l}' = \boldsymbol{K_L}\boldsymbol{n_c} = \begin{bmatrix} f_y & 0 & 0 \\ 0 & f_x & 0 \\ -f_y c_x & -f_x c_y & f_x f_y \end{bmatrix} \boldsymbol{n}_c = \begin{bmatrix} l_1 \\ l_2 \\ l_3 \end{bmatrix} \in R^3, \quad (10)$$

$$\boldsymbol{e}_l = \left[\frac{\boldsymbol{p_1}^T \boldsymbol{l}'}{\sqrt{{l_1}^2 + {l_2}^2}}, \frac{\boldsymbol{p_2}^T \boldsymbol{l}'}{\sqrt{{l_1}^2 + {l_2}^2}} \right]^T. \quad (11)$$

2) Optimization Function: The following are four different types of optimization errors (Fig.1(d)) in our factor graph:

$$\begin{aligned} \mathcal{T}^{i+1}, \mathcal{A}^{i+1} = & \arg\min_{\mathcal{T},\mathcal{A}} \sum_{k=1}^{K} \sum_{j=1}^{M} w_{kj}^i \boldsymbol{e}_{sl}^T \boldsymbol{\Omega}_{sl} \boldsymbol{e}_{sl} + \sum \boldsymbol{e}_p^T \boldsymbol{\Omega}_p \boldsymbol{e}_p \\ & + \sum \boldsymbol{e}_{ol}^T \boldsymbol{\Omega}_{ol} \boldsymbol{e}_{ol} + \sum \boldsymbol{e}_a^T \boldsymbol{\Omega}_a \boldsymbol{e}_a, \end{aligned} \quad (12)$$

where $\boldsymbol{e}_{sl}$ is the three-parameter line optimization error considering data association or temporary vanishing points, $\boldsymbol{e}_{ol}$ is the orthogonal representation error for structural lines not bound to a principal axis, $\boldsymbol{e}_a$ is the axis error that prevents

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

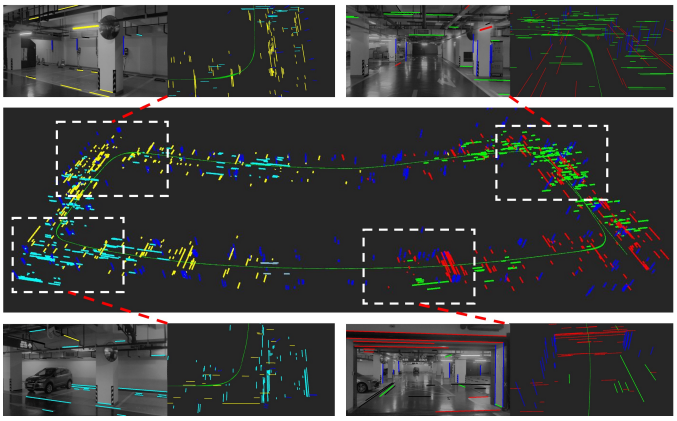

Fig. 4. Qualitative mapping results on the garage dataset. Lines in different colors represent different principal axes. Our system visually and intuitively illustrates the corresponding scene using structural line features and various colors.

excessive changes in the direction of the principal axis, and $\boldsymbol{e}_p$ is the point reprojection error. It is worth to notice that $\boldsymbol{e}_{sl}$ corresponds to the probability $\log p\left(\boldsymbol{L}_k \mid \boldsymbol{\mathcal{T}}, \boldsymbol{A}_j\right)$ with similar form in [50], which means the reprojection error that projecting 3D line $\boldsymbol{L}_k$ associated with axis $\boldsymbol{A}_j$ onto the frame that observed line $\boldsymbol{L}_k$.

3) Jacobian: We briefly show the chain rule of the reprojection error e_l concerning the three-parameter line $[\phi, \theta]^\top, r$. The computation results for pose variables follow the approach described in [32].

$$
\begin{aligned}
\frac{\partial e_l}{\partial\,[\phi,\theta]^\top} &= \frac{\partial e_l}{\partial \boldsymbol{v}} \frac{\partial \boldsymbol{v}}{\partial [\phi,\theta]^\top}, \frac{\partial e_l}{\partial r} = \frac{\partial e_l}{\partial \boldsymbol{L}_w} \frac{\partial \boldsymbol{L}_w}{\partial r}, \\
\frac{\partial \boldsymbol{v}}{\partial [\phi,\theta]^\top} &= \begin{bmatrix} \cos\phi\sin\theta & \sin\phi\cos\theta \\ \cos\phi\cos\theta & -\sin\phi\sin\theta \\ -\sin\phi & 0 \end{bmatrix}_{3\times 2}, \\
\frac{\partial \boldsymbol{L}_w}{\partial r} &= \begin{bmatrix} [\boldsymbol{v}]_\times \boldsymbol{R}_{ws} \boldsymbol{P}_s \\ \boldsymbol{0}_{3\times 1}^T \end{bmatrix}_{6\times 1}.
\end{aligned} \tag{13}
$$

IV. Experiments

A. Implementation Details

We run our system on a desktop PC with an Intel Core i7-12700 CPU. To evaluate the robustness of our system, we test it in complex outdoor datasets KITTI [54], [55] and Campus [23] and the challenging underground parking dataset (with glare and blur) BeVIS [56] and Garage [23]. Moreover, we generated a toy dataset to compare the performance of principal axis-anchored optimization with other line feature optimization methods.

To assess the tracking and mapping capabilities of our system, we compared it with open-sourced point-based SLAM systems ORB-SLAM3 [8], LDSO [57], and open-sourced point-line(-plane) based SLAM systems Structure-SLAM [46] (For outdoor environment, we use its point-line mode without the indoor structural constraint), Structure PLP-SLAM [39], and VPL-SLAM [23]. Considering that line features provide strong structural constraints, we also reduced the number of point features detected per frame of our system accordingly. We used the Absolute Trajectory Error Root Mean Square Error (ATE RMSE) as the metric for tracking evaluation. We evaluated all metrics using the average of 10 test runs for fairness.

B. Results

1) Results on Toy dataset: We generated multiple sets of lines with approximate directions and continuous poses and added perturbations to the initial values of the poses and line observations involved in the optimization. As shown in Tab.I, we compare the optimization time, the poses error, and the lines error with different representations (2-p [24], 4-p [19]) of the line under three scenarios. In the scenario where the poses are fixed as ground truth and do not participate in the optimization, the line error in our method is slightly larger than the orthogonal representation, as the orthogonal representation fully utilizes an independent line. Nevertheless, the optimization time of the 4-parameter representation is 4 times slower than our system.

When perturbations are added to the poses and included in the optimization, our method's pose and line errors are smaller than those of the other two representations because our method integrates the structural regularities into BA. Fig. 3 shows the qualitative convergence behavior of our method compared to the orthogonal representation [19] under given perturbations. Furthermore, in all scenarios, our optimization speed is the fastest. The results on the toy dataset demonstrate that our method achieves a better trade-off between accuracy and speed[1].

2) Results on Indoor Datasets: In structured indoor scenes characterized by low-texture regions and lighting variations, point-based SLAM systems perform poorly (especially methods based on optical flow), while line-feature systems provide structural information. Tab.II shows the tracking results for the indoor BeVIS [56] and Garage [23] datasets. Our system achieved state-of-the-art tracking results in low-texture and lighting-variable scenes. In these scenes, the AW constraint works well to make the line features in the same direction locally and globally (vertical lines are neat globally). Additionally, we qualitatively visualized the generated point-line maps (Fig. 4), where lines of different colors represent structural line features anchored to different principal axes. Our maps offer a clearer and more readable representation compared to other systems.

3) Results on Outdoor Datasets: We tested the tracking performance (Tab.II and Tab.III) of our system on outdoor datasets KITTI [54], [55] and Campus [23]. The outdoor scenes are complex, with dynamic objects and large loops. Following the classification in [23], we divided KITTI's 11 sequences into structured scenes with structured buildings and semi-structured scenes with trees, poles, and other objects with less geometric information and constraints than buildings. In structured scenes (part of KITTI and Campus [23]), our system effectively leveraged the scene's information, achieving good tracking metrics. In semi-structured scenes of KITTI,

[1]We modified the code at $github.com/HeYijia/vio\_data\_simulation$ and the comparison code from [24].

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

TABLE II
ATE RMSE (UNIT: M) RESULTS IN BEVIS, GARAGE AND CAMPUS DAATSETS

Method	BeVIS					Garage				Campus				
	00	01	02	03	Avg.	00	01	02	Avg.	00	01	02	03	Avg.
LDSO [57]	0.53*	1.49*	0.47*	0.18	0.67	1.43	1.20	1.02	1.22	0.67	1.74	2.21	5.10	2.43
ORB-SLAM3 [8]	0.41	0.04	0.29	0.26	0.25	1.75	2.08	1.14	1.66	1.18	1.61	2.56	4.37	2.43
Structure-SLAM [46]	0.33	-	12.64	0.13	-	-	-	-	-	1.32	1.88	3.84	-	-
Structure PLP-SLAM [39]	1.89*	**0.03**	-	0.18	-	3.19	6.73	1.13	3.68	2.40	5.67	6.37	20.77	8.80
VPL-SLAM [23]	0.37	0.16	0.41	0.16	0.28	0.85	1.05	1.01	0.97	0.65	0.87	2.12	3.62	1.82
Ours	**0.09**	0.04	**0.21**	**0.09**	**0.11**	**0.69**	**0.75**	**0.91**	**0.78**	**0.61**	**0.77**	**1.78**	**3.02**	**1.54**

*: The system crashed during operation, but some pose information was recorded before the crash.
-: The systems get lost during tracking because of visual degradation.

TABLE III
ATE RMSE (UNIT: M) RESULTS IN KITTI ODOMETRY DATASET

Method	Structured					Semi-structured					
	00	05	06	07	08	01	02	03	04	09	10
LDSO [57]	9.32	5.10	13.55	2.96	129.02	**11.68**	31.98	2.85	1.22	21.64	17.36
ORB3 [8]	8.07	6.71	15.19	2.89	55.97	-	25.38	1.05	1.25	8.04	8.76
Struct. [46]	6.62	12.62	23.67	3.36	104.92	-	23.62	2.68	1.22	13.78	**7.52**
PLP [39]	7.16	9.67	20.42	4.91	66.99	-	34.23	7.21	**0.47**	24.98	11.31
VPL [23]	6.04	4.64	**11.01**	1.64	66.05	-	23.23	0.81	0.75	8.22	8.49
Ours	**5.62**	**4.51**	12.47	**1.52**	**50.87**	-	**21.36**	**0.77**	0.72	**7.91**	8.75

TABLE IV
TIME ANALYSIS (UNIT: MS)

Method	Component		
	F. E.	Optim.	Track.
LDSO [57]	-	-	20
ORB3 [8]	11	126	19
Struct. [46]	38	55	42
PLP [39]	48	227	55
VPL [23]	21	114	32
Ours	21	127	33

our system can also effectively use the limited structural information, achieving relatively good results. It is worth noting that, apart from the direct method LDSO [57], other feature-based methods failed in KITTI 01. However, as shown in the indoor dataset, LDSO [57] performs poorly in areas with lighting changes.

C. Runtime Analysis

Except for LDSO [57], a direct method, all other methods are improvements on the ORB-SLAM [6] series. Therefore, we tested these methods in terms of feature extraction (F.E.), local mapping thread optimization (Optim.), and the time required to track (Track.) a single frame (Tab.IV). For LDSO, we only tested the time to track a single frame, as other modules cannot be fairly compared with ORB-SLAM [6] series. Additionally, Structure-SLAM [46] is mainly suitable for indoor datasets, which reduces some optimized parameters during local optimization. The results show that our method consumes less time than line-based systems, primarily due to our principal axis anchoring optimization approach and the deliberate reduction in point features.

D. Ablation Study

As shown in Table V, we conducted ablation experiments to evaluate the effectiveness of different strategies. For the hard association ablation, we replaced our proposed soft association algorithm with a hard association approach, where each line is anchored to only one principal axis. For the axis update ablation, we examined the impact of using or omitting the update strategies for the principal axes. For the vertical prior ablation, we assessed the effect of including or excluding the vertical prior. In the absence of vertical prior, all 37,800 proposals mentioned in Sec. III-C were tested. Lastly, for the vanishing point refinement ablation, we evaluated the effectiveness of applying or skipping the refinement for the two horizontal vanishing point directions. The results demonstrate the effectiveness of our strategies, particularly the vertical prior and axis update. The soft association strategy also improved accuracy, mainly by reducing the influence of mismatched lines.

TABLE V
ATE RMSE (UNIT: M) ABLATION OF DIFFERENT STRATEGIES

Method	Garage			KiTTi	
	s1	s2	s3	04	07
Hard Association	0.73	0.84	1.01	0.86	1.68
w/o. Axis Update	0.87	0.99	1.03	1.14	2.01
w/o. Vertical Prior	1.13	1.07	1.01	0.98	2.31
w/o. Vp Refinement	0.69	0.81	0.96	0.73	1.73
Our Full Model	0.69	0.75	0.91	0.72	1.52

V. CONCLUSION

We propose a SLAM system that optimizes line features with scene structural information. We improve efficiency and accuracy by anchoring co-directional lines to a principal axis and reducing parameters. Our principal axis management, combined with vertical priors and vanishing points, enhances robustness and minimizes mismatches. Experiments on multiple datasets confirm the system's reliability. We plan to extend our system to work with stereo, RGB-D, and IMU sensors in the future, building a more robust and versatile SLAM system.

REFERENCES

[1] Q. Chen, G. Li, X. Xue, and J. Pu, "Multi-lio: A lightweight multiple lidar-inertial odometry system," in *ICRA*. IEEE, 2024, pp. 13 748–13 754.

[2] C. Jiang, R. Gao, K. Shao, Y. Wang, R. Xiong, and Y. Zhang, "Li-gs: Gaussian splatting with lidar incorporated for accurate large-scale reconstruction," *IEEE Robotics and Automation Letters*, 2024.

[3] G. Li, Q. Chen, S. Hu, Y. Yan, and J. Pu, "Constrained gaussian splatting via implicit tsdf hash grid for dense rgb-d slam," *IEEE Transactions on Artificial Intelligence*, 2025.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

[4] A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse, "Monoslam: Real-time single camera slam," *TPAMI*, vol. 29, no. 6, pp. 1052–1067, 2007.

[5] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, "Orb-slam: a versatile and accurate monocular slam system," *IEEE TRO*, vol. 31, no. 5, pp. 1147–1163, 2015.

[6] R. Mur-Artal and J. D. Tardós, "Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras," *TRO*, vol. 33, no. 5, pp. 1255–1262, 2017.

[7] G. Li, Q. Chen, Y. Yan, and J. Pu, "Ec-slam: Effectively constrained neural rgb-d slam with tsdf hash encoding and joint optimization," *Pattern Recognition*, p. 112034, 2025.

[8] C. Campos, R. Elvira, J. J. G. Rodríguez, J. M. Montiel, and J. D. Tardós, "Orb-slam3: An accurate open-source library for visual, visual–inertial, and multimap slam," *TRO*, vol. 37, no. 6, pp. 1874–1890, 2021.

[9] A. Pumarola, A. Vakhitov, A. Agudo, A. Sanfeliu, and F. Moreno-Noguer, "Pl-slam: Real-time monocular visual slam with points and lines," in *ICRA*. IEEE, 2017, pp. 4503–4508.

[10] R. Gomez-Ojeda, F.-A. Moreno, D. Zuniga-Noël, D. Scaramuzza, and J. Gonzalez-Jimenez, "Pl-slam: A stereo slam system through the combination of points and line segments," *TRO*, vol. 35, no. 3, pp. 734–746, 2019.

[11] H. Li, Y. Xing, J. Zhao, J.-C. Bazin, Z. Liu, and Y.-H. Liu, "Leveraging structural regularity of atlanta world for monocular slam," in *ICRA*. IEEE, 2019, pp. 2412–2418.

[12] J. M. Coughlan and A. L. Yuille, "Manhattan world: Compass direction from a single image by bayesian inference," in *ICCV*, vol. 2, 1999, pp. 941–947.

[13] J. Straub, G. Rosman, O. Freifeld, J. J. Leonard, and J. W. Fisher, "A mixture of manhattan frames: Beyond the manhattan world," in *CVPR*, 2014, pp. 3770–3777.

[14] G. Schindler and F. Dellaert, "Atlanta world: An expectation maximization framework for simultaneous low-level edge grouping and camera calibration in complex man-made environments," in *CVPR*, vol. 1. IEEE, 2004, pp. I–I.

[15] H. Li, J. Zhao, J.-C. Bazin, P. Kim, K. Joo, Z. Zhao, and Y.-H. Liu, "Hong kong world: Leveraging structural regularity for line-based slam," *TPAMI*, vol. 45, no. 11, pp. 13 035–13 053, 2023.

[16] E. Eade and T. Drummond, "Edge landmarks in monocular slam," *IVC*, vol. 27, no. 5, pp. 588–596, 2009.

[17] J. Plucker, "Xvii. on a new geometry of space," *PTRSL*, no. 155, pp. 725–791, 1865.

[18] J. Sola, T. Vidal-Calleja, J. Civera, and J. M. M. Montiel, "Impact of landmark parametrization on monocular ekf-slam with points and lines," *IJCV*, vol. 97, pp. 339–368, 2012.

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

[20] L. Zhang and R. Koch, "Structure and motion from line correspondences: Representation, projection, initialization and sparse bundle adjustment," *JVCIR*, vol. 25, no. 5, pp. 904–915, 2014.

[21] K. S. Roberts, "A new representation for a line," in *CVPR*. IEEE, 1988, pp. 635–636.

[22] H. Lim, J. Jeon, and H. Myung, "Uv-slam: Unconstrained line-based slam using vanishing points for structural mapping," *RAL*, vol. 7, no. 2, pp. 1518–1525, 2022.

[23] Q. Chen, Y. Cao, J. Hou, G. Li, S. Qiu, B. Chen, X. Xue, H. Lu, and J. Pu, "Vpl-slam: A vertical line supported point line monocular slam system," *TITS*, 2024.

[24] B. Xu, P. Wang, Y. He, Y. Chen, Y. Chen, and M. Zhou, "Leveraging structural information to improve point line visual-inertial odometry," *RAL*, vol. 7, no. 2, pp. 3483–3490, 2022.

[25] W. Zhao, H. Sun, X. Zhang, and Y. Xiong, "Visual slam combining lines and structural regularities: Towards robust localization," *TIV*, 2023.

[26] W. Chen, G. Shang, A. Ji, C. Zhou, X. Wang, C. Xu, Z. Li, and K. Hu, "An overview on visual slam: From tradition to semantic," *RS*, vol. 14, no. 13, p. 3010, 2022.

[27] G. Klein and D. Murray, "Parallel tracking and mapping for small ar workspaces," in *ISMAR*. IEEE, 2007, pp. 225–234.

[28] J. Engel, T. Schöps, and D. Cremers, "Lsd-slam: Large-scale direct monocular slam," in *ECCV*. Springer, 2014, pp. 834–849.

[29] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "Orb: An efficient alternative to sift or surf," in *ICCV*. IEEE, 2011, pp. 2564–2571.

[30] D. Gálvez-López and J. D. Tardos, "Bags of binary words for fast place recognition in image sequences," *TRO*, vol. 28, no. 5, pp. 1188–1197, 2012.

[31] S. Maity, A. Saha, and B. Bhowmick, "Edge slam: Edge points based monocular visual slam," in *ICCV Workshop*. IEEE, 2017, pp. 2408–2417.

[32] X. Zuo, X. Xie, Y. Liu, and G. Huang, "Robust visual slam with point and line features," in *IROS*. IEEE/RSJ, 2017, pp. 1775–1782.

[33] Y. He, J. Zhao, Y. Guo, W. He, and K. Yuan, "Pl-vio: Tightly-coupled monocular visual–inertial odometry using point and line features," *Sensors*, vol. 18, no. 4, p. 1159, 2018.

[34] C. Zhang, Z. Fang, X. Luo, and W. Liu, "Accurate and robust visual slam with a novel ray-to-ray line measurement model," *IVC*, vol. 140, p. 104837, 2023.

[35] J. Yan, Y. Zheng, J. Yang, L. Mihaylova, W. Yuan, and F. Gu, "Plpf-vslam: An indoor visual slam with adaptive fusion of point-line-plane features," *JFR*, vol. 41, no. 1, pp. 50–67, 2024.

[36] K. Xu, Y. Hao, S. Yuan, C. Wang, and L. Xie, "Airslam: An efficient and illumination-robust point-line visual slam system," *TRO*, 2025.

[37] S. J. Lee and S. S. Hwang, "Elaborate monocular point and line slam with robust initialization," in *ICCV*. IEEE, 2019, pp. 1121–1129.

[38] H. Wei, F. Tang, Z. Xu, and Y. Wu, "Structural regularity aided visual-inertial odometry with novel coordinate alignment and line triangulation," *RAL*, vol. 7, no. 4, pp. 10 613–10 620, 2022.

[39] F. Shu, J. Wang, A. Pagani, and D. Stricker, "Structure plp-slam: Efficient sparse mapping and localization using point, line and plane for monocular, rgb-d and stereo cameras," in *ICRA*. IEEE, 2023, pp. 2105–2112.

[40] P. Smith, I. Reid, and A. J. Davison, "Real-time monocular slam with straight lines." in *BMVC*, vol. 6, 2006, pp. 17–26.

[41] L. Zhou, G. Huang, Y. Mao, S. Wang, and M. Kaess, "Edplvo: Efficient direct point-line visual odometry," in *ICRA*. IEEE, 2022, pp. 7559–7565.

[42] H. Zhou, D. Zou, L. Pei, R. Ying, P. Liu, and W. Yu, "Structslam: Visual slam with building structure lines," *TVT*, vol. 64, no. 4, pp. 1364–1375, 2015.

[43] P. Kim, B. Coltin, and H. J. Kim, "Visual odometry with drift-free rotation estimation using indoor scene regularities." in *BMVC*, vol. 2, no. 6, 2017, p. 7.

[44] Kim, Pyojin and Coltin, Brian and Kim, H Jin, "Low-drift visual odometry in structured environments by decoupling rotational and translational motion," in *ICRA*. IEEE, 2018, pp. 7247–7253.

[45] H. Li, J. Yao, J.-C. Bazin, X. Lu, Y. Xing, and K. Liu, "A monocular slam system leveraging structural regularity in manhattan world," in *ICRA*. IEEE, 2018, pp. 2518–2525.

[46] Y. Li, N. Brasch, Y. Wang, N. Navab, and F. Tombari, "Structure-slam: Low-drift monocular slam in indoor environments," *RAL*, vol. 5, no. 4, pp. 6583–6590, 2020.

[47] J. Liu and Z. Meng, "Visual slam with drift-free rotation estimation in manhattan world," *RAL*, vol. 5, no. 4, pp. 6512–6519, 2020.

[48] Z. Zhou, Z. Gao, and J. Xu, "Tracking by detection: Robust indoor rgb-d odometry leveraging key local manhattan world," *RAL*, 2024.

[49] Y. Cheng, "Mean shift, mode seeking, and clustering," *IEEE TPAMI*, vol. 17, no. 8, pp. 790–799, 1995.

[50] S. L. Bowman, N. Atanasov, K. Daniilidis, and G. J. Pappas, "Probabilistic data association for semantic slam," in *ICRA*. IEEE, 2017, pp. 1722–1729.

[51] C. Akinlar and C. Topal, "Edlines: A real-time line segment detector with a false detection control," *PRL*, vol. 32, no. 13, pp. 1633–1642, 2011.

[52] R. G. Von Gioi, J. Jakubowicz, J.-M. Morel, and G. Randall, "Lsd: A fast line segment detector with a false detection control," *TPAMI*, vol. 32, no. 4, pp. 722–732, 2008.

[53] X. Lu, J. Yaoy, H. Li, Y. Liu, and X. Zhang, "2-line exhaustive searching for real-time vanishing point estimation in manhattan world," in *IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, 2017, pp. 345–353.

[54] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset," *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1231–1237, 2013.

[55] Y. Liao, J. Xie, and A. Geiger, "Kitti-360: A novel dataset and benchmarks for urban scene understanding in 2d and 3d," *TPAMI*, vol. 45, no. 3, pp. 3292–3310, 2022.

[56] X. Shao, Y. Shen, L. Zhang, S. Zhao, D. Zhu, and Y. Zhou, "Slam for indoor parking: A comprehensive benchmark dataset and a tightly coupled semantic framework," *ACM Transactions on Multimedia Computing, Communications and Applications*, vol. 19, no. 1, pp. 1–23, 2023.

[57] X. Gao, R. Wang, N. Demmel, and D. Cremers, "Ldso: Direct sparse odometry with loop closure," in *IEEE/RSJ IROS*. IEEE, 2018, pp. 2198–2204.