Leveraging Structural Information to Improve Point Line Visual-Inertial Odometry

Bo Xu ⬤, Peng Wang, Yijia He ⬤, Yu Chen, Yongnan Chen, and Ming Zhou

Abstract—Leveraging line features to improve the accuracy of the SLAM system has been studied in many works. However, making full use of the characteristics of different line features (parallel, non-parallel) to improve the SLAM system is rarely mentioned. In this paper, we designed a VIO system based on points and straight lines, which divides straight lines into structural (that is, straight lines parallel to each other) and non-structural. To optimize the line features effectively, we used two-parameter representation methods for both structural and non-structural straight lines. Furthermore, we designed a stable line matching method based on frame-to-frame (2D-2D) and frame-to-map (2D-3D) strategies which can significantly improve the trajectory accuracy of the system. We conducted ablation experiments on synthetic data and public datasets, and also compared our method with state-of-the-art algorithms. The experiments verified the combination of different line features can improve the accuracy of the VIO system, and also demonstrated the effectiveness of our system.

Index Terms—SLAM, visual inertial odometry, structural information.

I. INTRODUCTION

S IMULTANEOUS motion estimating and mapping is widely used in the field of intelligent robots, such as autonomous driving, rescue, and augmented reality. With camera and inertial measurement unit (IMU) being low-cost and efficient sensors, the visual-inertial odometry system (VIO) can overcome the shortcomings of the two sensors and improve the accuracy and robustness of localization. The existing VIO systems mainly use points as visual features to estimate ego-pose and build a sparse map of 3D points [1]–[3]. However, in some textureless or illumination challenging environments, point-based VIOs may fail in pose estimation [4].

Bo Xu and Yu Chen are with the School of Geodesy and Geomatics, Wuhan University, Wuhan 430079, China (e-mail: boxu1995@whu.edu.cn; ychen7121@njtech.edu.cn).

Peng Wang, Yongnan Chen, and Ming Zhou are with the Faculty of Robot Science and Engineering, Northeastern University, Shenyang 110819, China (e-mail: 1801944@stu.neu.edu.cn; ynchen96@gmail.com; zhouminganhui@qq.com).

Yijia He is with the Kuaishou Technology, Beijing 100080, China (e-mail: heyijia2016@gmail.com).

Line features exist widespread in man-made environments, which can provide additional visual constraints and build a map with richer information for automatic navigation. Thus, VIO systems based on point and line features have attracted widespread attention. Current line features can be divided into two categories based on the type of line features: non-structural lines [5]–[8], and structural lines [9]–[12]. The non-structural lines have more universality and robustness because they can be operated in different kinds of environments, but non-structural lines have no effective directional constraints as structural lines do. Structural lines can be found in man-made environments, which are abstracted as a set of blocks sharing three common dominant directions, known as Manhattan world [13], they encode the global orientations of the local environments, these constraints can reduce the drift of the yaw angle, however, when the environment is complex, the constraints are not effective for pose estimation.

In order to overcome the shortcomings of insufficient visual information in the point-based VIO system, the line features are a good supplement. PL-VIO proposed by He *et al.* [7] integrates straight lines into the VIO system based on point features. In this work, the straight lines are represented with Plücker coordinates, and a minimal four-parameter orthonormal representation is used for optimization because the Plücker coordinates are over-parameterized. The Trifo-VIO system proposed by Zheng *et al.* [14] uses Kalman filter to integrate the straight lines into the VIO system. The authors use geometric constraints expressed by the normal vector of line to construct the constraints.

However, none of them distinguish the difference between structural and non-structural lines, and the parallel constraints between structural lines are not adopted. Camposeco *et al.* [15] incorporated vanishing points into the VIO system, using global constraint information of vanishing points to correct the yaw angle drift in pose estimation. However, the system only regards structural lines as intermediate results and does not make full use of structural lines to correct translation. Zou *et al.* [16] proposed a new parameter form of structural lines and integrated the structural line features into the VIO system, but non-structural line features are not employed.

Overall, non-structural lines and structural lines have the potential to improve the performance of the VIO system with different advantages, by leveraging both of them, the estimator will be more robust and accurate in the complex environment. This paper proposes a tightly coupled monocular VIO system named PLS-VIO (Point, Non-structural line, Structural line VIO), including visual point features, non-structural line
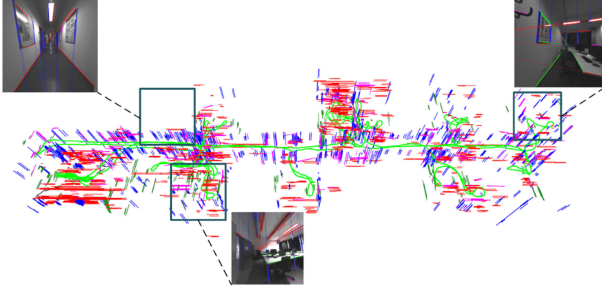
Fig. 1.   The proposed monocular VIO system that builds line map. Red, green and blue lines in the map are the landmarks of structrual line feature in X, Y, Z direction, the purple lines are the landmarks of non-structural line feature. Three sub-images show the lines detected in the image.
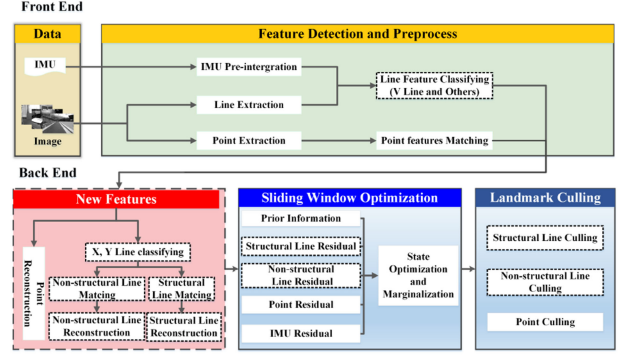

Fig. 2.   Overview of our PLS-VIO system. The Front-End module is used to extract information from the raw measurement; Line classification and state variables are estimated with sliding window optimization in the Back-End.
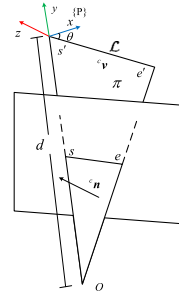

Fig. 3.   the endpoint $\mathbf{s}'$ and $\mathbf{e}'$ of 3D line $\mathcal{L}$ are projected to the image plane to form $\mathbf{s}$ and $\mathbf{e}$. $\mathbf{s}'$, $\mathbf{e}'$ and optical center O of camera frame construct the plane π, the direction vector of $\mathcal{L}$ is $^c\mathbf{v}$, the normal vector of π is $^c\mathbf{n}$, the distance from O to $\mathbf{s}'$ is d, we define local coordinate system $\{P\}$ of which the origin is $\mathbf{s}'$, the angle between the $^c\mathbf{v}$ expressed in the $\{P\}$ and x axis of the $\{P\}$ is θ.

features, and structural line features to achieve accurate pose estimation and point-line map construction, as shown in Fig. 1. The main contributions of this work include:

- We divide lines observed by the VIO system into structural lines and non-structural lines, compared with the single type of line features, our method utilizes the different line constraints to improve the accuracy and robustness of pose estimation and mapping in the complex environment.
- We design a 2D-2D and 2D-3D line matching algorithm to reduce mismatching and tracking lost in a long period of line features. In order to optimize the line features efficiently, we introduce a 2-parameter representation of non-structural lines, the line features are fused efficiently into the optimization-based estimator.
- We provide ablation experiments to verify the effectiveness of the proposed algorithm on the synthetic data, EuRoc dataset [17], and TUM VI benchmark [18]. Experimental results demonstrate that our system is capable of accurate pose estimation and mapping.

II. SYSTEM OVERVIEW

The system proposed in this paper is based on VINS-Mono [2]. VINS-Mono uses the optimization method to tightly couple the IMU observation and the visual observation of point features. Our system adds non-structural lines and structural lines, as well as constructs corresponding constraints.

As shown in Fig. 2, our system contains two modules: the front end and the back end. In the front end, raw measurements of IMU and image are pre-processed, including IMU pre-integration, point detection and matching, line detection and vertical line classification.

In the back end, the operations for the non-structural lines and structural lines are mainly introduced. We pass the vertical line which is aligned with gravity direction and non-vertical lines to the back end, X, Y direction and non-structural lines from non-vertical direction lines are further classified, this will be described in III-B. Nextly, two different line matching strategies are operated, for the efficiency of the code operation and the simplification of the line data management, we move the matching of lines to the back end, this process will be described in III-C. After this, we initialize the lines to get 3D line landmarks,

which will be introduced in the III-D. Finally, the IMU body state and 3D landmarks in the map will be optimized by minimizing the sum of the IMU residual, prior residual, point re-projection residual, and line re-projection residual, all these residuals will be introduced in the IV.

III. STRUCTURAL AND NON-STRUCTURAL LINE METHODOLOGY

In this section, we introduce the implementation details of non-structural lines and structural lines in the VIO system. First, the parameter representation of point and line landmarks is described. Second, we introduce the classification of the lines. Finally, the different matching methods of lines and the initialization for non-structural lines and structural lines are presented.

A. Landmarks Representation

1) Point Representation: We use the inverse depth $\lambda \in \mathbb{R}$ to parameterize the point landmark from the first keyframe in which it is observed. Given the point observation $\mathbf{z} = [u, v, 1]^T$ in the normalized image plane, the 3D position of landmark is obtained by $\mathbf{f} = \frac{1}{\lambda} \cdot \mathbf{z}$.

2) Non-Structural Line Representation: For the non-structural line, as shown in Fig. 3, the plane π is composed of the two endpoints $\mathbf{s}' \in \mathbb{R}^3$ and $\mathbf{e}' \in \mathbb{R}^3$ of the 3D line $\mathcal{L}$ and the optical center O of the camera, the 3D line $\mathcal{L}$ can be
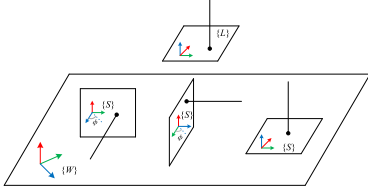
Fig. 4. Start frames $\{S\}$ of X, Y, Z direction structural lines from the parameter space $\{L\}$, the origin of start frame is in the world coordinate system $\{W\}$, the orientation of start frame is rotated about ϕ from the world coordinate system.

expressed by $\mathcal{L} = [{}^c\mathbf{n}^{\mathrm{T}}, {}^c\mathbf{v}^{\mathrm{T}}]^{\mathrm{T}}$, where ${}^c\mathbf{n} \in \mathbb{R}^3$ is the normal vector of π, ${}^c\mathbf{v} \in \mathbb{R}^3$ is the direction vector of $\mathcal{L}$. Due to the noise of the host frame that line is anchored is small, we fix the normal vector ${}^c\mathbf{n}$ of the plane where the line lies, and use the parameters of 2 DOF to represent the 4 DOF line. The local coordinate system on the plane π is defined as $\{P\}$, to simplify the representation of $\mathcal{L}$, we let the origin of $\{P\}$ be $\mathbf{s}'$, then let the direction of y axis be aligned to the ray passing from O to $\mathbf{s}'$ and let direction of z axis be parallel to the normal vector ${}^c\mathbf{n}$ of plane π. Due to the orthogonality of the coordinate axes, the x axis is perpendicular to the y axis and the z axis. The distance from O to $\mathbf{s}'$ is $d \in \mathbb{R}$.

To reduce the number of line parameters during optimization, we propose to use a compact parameterization that has only two parameters: θ and $\rho = 1/d$. The θ is the angle between line direction ${}^c\mathbf{v}'$ in local coordinate $\{P\}$ and the x axis of the $\{P\}$. The ${}^c\mathbf{v}'$ can be obtained by ${}^c\mathbf{v}' = \mathbf{R}_C^P {}^c\mathbf{v}$, where $\mathbf{R}_C^P \in \mathbb{R}^{3\times3}$ is the rotation matrix of camera coordinate w.r.t the local coordinate $\{P\}$.

3) Structural Line Representation: We represent structural lines in multiple local Manhattan worlds with different orientations [16]. Specifically, as shown in Fig. 4, each structural line is anchored to the local coordinate system where it is first observed, and we define this anchored coordinate system as a start frame $\{S\}$. The rotation and translation of start frame $\{S\}$ w.r.t the world coordinate is $(\mathbf{R}_S^W, \mathbf{P}_S^W)$. The $\mathbf{R}_S^W$ is the rotation matrix of associated local Manhattan world frame w.r.t world frame, which is rotated about ϕ from the world coordinate system. The $\mathbf{P}_S^W$ is same as the position of camera coordinate w.r.t world frame in which the line is firstly observed.

In order to express uniformly the X, Y, Z direction lines in $\{S\}$, we define again a parameter space $\{L\}$, the transformation from the parameter space $\{L\}$ to the start frame $\{S\}$ is a pure rotation $\mathbf{R}_L^S$. In the parameter space, the structural line can be represented as the intersection point $\mathbf{l}_p^l = [a, b, 0]^{\mathrm{T}}$ on the XY plane, we use the inverse depth to represent the intersection point, namely $[\theta, \rho, 0]^{\mathrm{T}}$, where $\rho = \sqrt{a^2 + b^2}$ and $\theta = atan2(b, a)$, we adopt this representation to speed up the convergence of structural line.

B. Structural Line and Non-Structural Line Classification

We use the LSD algorithm [19] to detect lines in the image and then classify these straight lines into structural lines and non-structural lines. The vanishing points [15] in the image are used to recognize the structural lines. We detect the vanishing point of

Z direction with the aid of IMU information [16], for structural lines of Z direction, we draw a ray from the vanishing point $\mathbf{v}_z$ of Z direction to the middle point of line segment S. The angle A_{err} and distance D_{err} between the ray and S are further calculated. If A_{err} is less than the angle threshold A_{th}, D_{err} is less than the distance threshold D_{th}, we consider S to be the structural line of Z direction. For the remaining lines, we first attach the line to the detected Manhattan world according to the consistency of angle and distance, when the number of unclassified lines is greater than 60% of the total number of detected lines, we use the RANSAC method [15] to detect the vanishing points in X and Y directions, and then classify lines into the new Manhattan world similar to the method of Z direction line classification, finally, the lines that are not classified are considered as non-structural lines.

C. Line Matching for 2D-2D and 2D-3D

To improve the stability and accuracy of line tracking, we combine two tracking strategies, namely frame-to-frame line tracking and frame-to-map line tracking. In general, we perform frame-to-frame tracking method to track new detected lines, if the number of matched lines is too small, we will perform the frame-to-map method to increase the number of the matched lines. Compared with the matching method in [16], the main aim of performing firstly frame-to-frame matching method is to avoid mismatching due to inaccurate position estimation of 3D line landmarks.

1) Frame-to-Frame Line Tracking Method: For the frame-to-frame line tracking method, we sample all lines in the previous frame and get the set of sampling points $p_i \in \{P\_sample\}$, and then use the epipolar searching method [20] to find corresponding candidate matching points in the current frame. The ZMSSD (Zero-mean Sum of Squared Differences) template [21] is used to calculate the matching score of two points. We select the candidate point with the highest matching score as the tracked point and get the set of tracked points $p_i' \in \{P\_tracked\}$. The tracked point is valid if the distance from it to the line of the current frame is less than the threshold m_th, m_th is set to be 5 pixels in our implementation. We consider the line to be the best matching if the number of valid points is greater than 0.8 times of sampling points on the line of the previous frame.

2) Frame-to-Map Line Tracking Method: For the frame-to-map line tracking, we use the ZNCC (Zero-normalized cross-correlation) matching method [22]. For a 3D line, we get the latest observation frame F_i in the history frames corresponding to the line, the lines between F_i and the current frame are matched using ZNCC method. Due to the rapid movement of the camera and the occlusion in the scene, the lengths of lines to be matched are quite different, which affects the accuracy of the matching. We use epipolar geometric constraints to determine the sampling range of the line to assist ZNCC matching, that is, after calculating essential matrix $\mathbf{E}$ between F_i and the current frame, the line's endpoints in F_i are projected to the current frame according to the epipolar geometry, the corresponding sampling range is determined by intersecting two epipolar lines with matching line in the current frame, which improves the success rate of ZNCC matching method.

D. Initialization for Structural Line and Non-Structure Line

The stability and accuracy of the line initialization have great effects on the pose estimation. For the two parameterized expressions of structural lines and non-structural lines, we use different initialization methods to determine reasonable initial values.

1) Initialization of Non-Structural Line: The line segment in the normalized image plane can be represented by two endpoints $\mathbf{s}^{c_1} = [u_s, v_s, 1]^{\mathrm{T}}$ and $\mathbf{e}^{c_1} = [u_e, v_e, 1]^{\mathrm{T}}$. Three non-collinear points, including two endpoints of a line segment and the optical center O of camera, determine a plane $\boldsymbol{\pi} = [\pi_x, \pi_y, \pi_z, \pi_w]^{\mathrm{T}}$, given the two plane $\boldsymbol{\pi_1}$ and $\boldsymbol{\pi_2}$ in the camera frame c_1, the dual Plücker matrix $^c\mathcal{L}^*$ can be computed by:

$$^c\mathcal{L}^* = \begin{bmatrix} [^c\mathbf{v}]_\times & ^c\mathbf{n} \\ -^c\boldsymbol{n}^{\mathrm{T}} & 0 \end{bmatrix} = \boldsymbol{\pi_1}\boldsymbol{\pi_2}^{\mathrm{T}} - \boldsymbol{\pi_2}\boldsymbol{\pi_1}^{\mathrm{T}} \in \mathbb{R}^{4\times 4} \quad (1)$$

where $[\cdot]_\times$ is the skew-symmetric matrix of a three-dimensional vector.

We get the Plücker coordinate of the line $^c\mathcal{L} = [^c\mathbf{n}^{\mathrm{T}}, {}^c\mathbf{v}^{\mathrm{T}}]^{\mathrm{T}}$ from the dual Plücker matrix, where $^c\mathbf{n} \in \mathbb{R}^3$ denotes the normal vector of the plane determined by $^c\mathcal{L}$ and the origin of the camera frame c_1. The $^c\mathbf{v} \in \mathbb{R}^3$ denotes the direction vector determined by the two endpoints of $^c\mathcal{L}$. We transform $^c\mathbf{v}$ in the camera coordinate into local coordinate $\{P\}$ by $^c\mathbf{v}' = R_C^P{}^c\mathbf{v}$, where $R_C^P = R_P^{C\mathrm{T}}$, the columns of R_P^C is composed of the x, y, z axis of local coordinate, respectively. Therefore, we can initialize the θ as the angle between the x axis direction of the local coordinate system and $^c\mathbf{v}'$, and ρ is the expression of the inverse depth, and is generally initialized as $\rho_0 = 0.2$.

2) Initialization of Structural Line: The initialization of the structural line also needs to first calculate the Plücker coordinate of the line $^c\mathcal{L} = [^c\mathbf{n}^{\mathrm{T}}, {}^c\mathbf{v}^{\mathrm{T}}]^{\mathrm{T}}$. And then we get the three-dimensional endpoints expression of line in the world coordinate $^w\mathcal{L} = [\mathbf{s}^{w\mathrm{T}}, \mathbf{e}^{w\mathrm{T}}]^{\mathrm{T}}$ using line trimming [23]. To obtain the intersection point of $^w\mathcal{L}$ and XY plane in the world coordinate system, we transfer $^l\mathbf{p}$ to the world coordinate system to get $^w\mathbf{p}$ through formula (2):

$$^w\mathbf{p} = \begin{bmatrix} \mathbf{R_S^W R_L^S} & \mathbf{P}_S^W \\ \mathbf{0} & 1 \end{bmatrix}^{-\mathrm{T}} {}^l\mathbf{p} \quad (2)$$

Then we intersect $^w\mathcal{L}$ with the plane $^w\mathbf{p}$ to get the intersection point $^w\mathbf{l}_p$, and transfer $^w\mathbf{l}_p$ to the parameter space to get point $^l\mathbf{l}_p$ in return.

$$^l\mathbf{l}_p = \mathbf{R}_L^{S\,\mathrm{T}}\mathbf{R}_S^{W\,\mathrm{T}}(^w\mathbf{l}_p - \mathbf{P}_S^W) \quad (3)$$

We use the intersection $^l\mathbf{l}_p = [^l l_{px}, {}^l l_{py}, 0]^{\mathrm{T}}$ to initialize the structural line. θ can be initialized as $\theta_0 = atan2(^l l_{py}, {}^l l_{px})$. The inverse depth can be initialized as $\rho_0 = 1/\sqrt{^l l_{px}{}^2 + {}^l l_{py}{}^2}$.

IV. VIO WITH LINE AND POINT

In this section, we will fuse the IMU and visual information with the sliding window optimization to build VIO system which estimates body states and 3D landmarks.

A. VIO System Formulation

We optimize all the state variables in the sliding window by minimizing the sum of cost terms from IMU residual, visual residual and prior residual:

$$\boldsymbol{\chi} = \arg\min_{\boldsymbol{\chi}} \|\mathbf{r}_p - \mathbf{J}_p\boldsymbol{\chi}\|^2 + \sum_{i\in\mathcal{B}} \rho\left(\|\mathbf{r}_b\|^2_{\sum_{b,b_i+1}}\right)$$
$$+ \sum_{(i,k)\in\mathcal{F}} \rho\left(\left\|\mathbf{r}_{f_k}^{c_i}\right\|^2_{\sum_\mathcal{F}}\right) + \sum_{(i,l)\in\mathcal{L}} \rho\left(\left\|\mathbf{r}_{L_l}^{c_i}\right\|^2_{\sum_\mathcal{L}}\right)$$
$$+ \sum_{(i,s)\in\mathcal{C}} \rho\left(\left\|\mathbf{r}_{c_s}^{c_i}\right\|^2_{\sum_\mathcal{C}}\right) \quad (4)$$

Where $\mathbf{r}_b$ is the IMU measurement residual, $\mathbf{r}_{f_k}^{c_i}$ is the re-projection residual of point, $\mathbf{r}_{L_l}^{c_i}$ and $\mathbf{r}_{c_s}^{c_i}$ are the re-projection of non-structural line and structural line respectively. $\mathbf{r}_p$ and $\mathbf{J}_p$ are the prior residual and Jacobian from marginalization operator [2], respectively. $\rho(\cdot)$ is the Cauchy robust function used to suppress outliers. $\sum_{(\cdot)}$ is the covariance matrix of a measurement. The covariance matrix $\sum_{b,b_i+1}$ of IMU is calculated by covariance propagation with IMU measurement noise, the covariance matrix of visual measurement is determined by the prior knowledge.

B. Point Feature Measurement Model

For point features, we use the distance between the projected point and the observed point in the normalized image plane defined as re-projection error to represent the residual. Given the kth point feature measurement at frame c_j, $\mathbf{z}_{f_k}^{c_j} = \left[u_{f_k}^{c_j}, v_{f_k}^{c_j}, 1\right]^{\mathrm{T}}$, the re-projection error is defined as:

$$\mathbf{r}_{f_k}^{c_i} = \begin{bmatrix} \frac{x^{c_j}}{z^{c_j}} - u_{f_k}^{c_j} \\ \frac{y^{c_j}}{z^{c_j}} - v_{f_k}^{c_j} \end{bmatrix} \quad (5)$$

where $[x^{c_j}, y^{c_j}, z^{c_j}]^{\mathrm{T}}$ is the projected point from the first observation frame of the feature.

C. Non-Structural Line Measurement Model

For the non-structural line, we express the measurement model by transferring the line segment parameters observed in the first keyframe to the other keyframe which observes the line. The direction vector of line in the local coordinate can be calculated by $^d\mathbf{v} = [cos\theta, sin\theta, 0]^{\mathrm{T}}$, we can transfer the $^d\mathbf{v}$ from the local coordinate to the camera coordinate system to get the line direction $^c\mathbf{v}$.

$$^c\mathbf{v} = \mathbf{R}_P^C{}^d\mathbf{v} \quad (6)$$

Where $\mathbf{R}_P^C$ is the rotation matrix from local coordinate to the camera frame coordinate, the columns of $\mathbf{R}_P^C$ is composed of the x, y, z axis of local coordinate $\{P\}$, respectively.

The one endpoint $\mathbf{s}$ of the line segment in the camera frame can be calculated by

$$\mathbf{s} = \frac{\mathbf{y}}{\|\mathbf{y}\|} \cdot d \quad (7)$$

Where $\mathbf{y}$ is the y axis of the local coordinate.

To obtain the projection of a line on the normalized image plane, it requires to transfer both the endpoint $\mathbf{s}$ and the direction $^c\mathbf{v}$ in the first observation keyframe to the target keyframe. We get the endpoint $\mathbf{s}'$ and direction $^c\mathbf{v}'$ in the target frame by:

$$\mathbf{s}' = \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \mathbf{R}_{C_j}^{W\,\mathrm{T}}(\mathbf{R}_{C_i}^{W}\mathbf{s} + \mathbf{P}_{C_i}^{W}) - \mathbf{R}_{C_j}^{W\,\mathrm{T}}\mathbf{P}_{C_j}^{W} \quad (8)$$

$$^c\mathbf{v}' = \mathbf{R}_{C_j}^{W\,\mathrm{T}}\mathbf{R}_{C_i}^{W}{}^c\mathbf{v} \quad (9)$$

where $(\mathbf{R}_{C_i}^{W}, \mathbf{P}_{C_i}^{W})$ is the pose of first keyframe that observes line feature, and $(\mathbf{R}_{C_j}^{W}, \mathbf{P}_{C_j}^{W})$ is the pose of target keyframe that observes line features, we absorb the transformation from IMU to the camera.

We get the line equation on the target normalized image plane by

$$\mathbf{l}_l^{m_i} = [\mathbf{s}']_\times {}^c\mathbf{v}' \quad (10)$$

The measurement of the line segment $\mathbf{z}_{L_l}^{m_i}$ on the normalized image plane consists with two endpoints $\mathbf{s}_l^{m_i} = [u_s, v_s, 1]^{\mathrm{T}}$ and $\mathbf{e}_l^{m_i} = [u_e, v_e, 1]^{\mathrm{T}}$, the line re-projection residual is defined as:

$$\mathbf{r}_{L_l}^{c_i} = \frac{1}{\|\mathbf{z}_{L_l}^{m_i}\|} \begin{bmatrix} d(\mathbf{s}_l^{m_i}, \mathbf{l}_l^{m_i}) \\ d(\mathbf{e}_l^{m_i}, \mathbf{l}_l^{m_i}) \end{bmatrix} \quad (11)$$

With $d(\mathbf{s}, \mathbf{l})$ is the distance from the endpoint $\mathbf{s}$ to the projection line $\mathbf{l}$:

D. Structural Line Measurement Model

The residual form of the structural line is to transfer the line parameter in the parameter space $\{L\}$ to the start frame $\{S\}$, and then to the other keyframe that observe the 3D line. In the target keyframe, the line re-projection residual is constructed.

Using (12), the parameters of the line are transferred from the parameter space of the first observation keyframe to the camera coordinate of the target keyframe, in order to simplify the formula, we absorb the transformation from the IMU to the camera.

$$^c\mathbf{l}_p = \mathbf{R}_W^C\mathbf{R}_S^W\mathbf{R}_L^{Sl}\mathbf{l}_p + (\mathbf{R}_W^C\mathbf{P}_S^W + \mathbf{P}_W^C) \quad (12)$$

where $(\mathbf{R}_S^W, \mathbf{P}_S^W)$ represents the transformation from the start frame to the world coordinate, $(\mathbf{R}_W^C, \mathbf{P}_W^C)$ represents the transformation from the world coordinate to the target keyframe. $^l\mathbf{l}_p = [a, b, 0]^{\mathrm{T}}$, a, b are expressed by the line parameters ρ, θ, with $a = \frac{\cos\theta}{\rho}$, $b = \frac{\sin\theta}{\rho}$.

The direction of structural line is represented in the parameter space $^l\mathbf{v} = [0, 0, 1]^{\mathrm{T}}$, and we use (13) to transform it into the camera frame coordinate.

$$^c\mathbf{v} = \mathbf{R}_W^C\mathbf{R}_S^W\mathbf{R}_L^{Sl}\mathbf{v} \quad (13)$$

We get the line equation on the target frame by:

$$\mathbf{l}_s^{m_i} = [^c\mathbf{l}_p]_\times {}^c\mathbf{v} \quad (14)$$

Similar to the non-structural line, the structural line re-projection residual is also defined by the distance between the observation and the line from perspective projection.
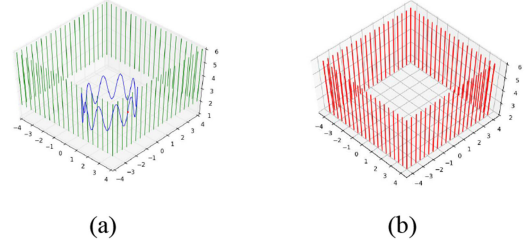

Fig. 5. Synthetic data and reconstruction of 3D lines by our method. (a) groundtruth. (b) our method.

TABLE I
COMPARISON OF RECONSTRUCTION ACCURACY[CM] / TIME[S] OF DIFFERENT LINE REPRESENTATION IN DIFFERENT PIXEL NOISE LEVEL

Para	0 pixel		0.5 pixel		1.0 pixel		1.5 pixel	
	trans.	time.	trans.	time.	trans.	time.	trans.	time.
2-para (ours)	0.06	18.54	2.29	18.06	5.13	17.85	7.54	18.19
2-para ([30])	0.06	18.05	2.90	18.18	5.95	18.13	7.61	18.38
4-para ([7])	0.05	36.89	1.80	36.08	3.65	36.48	5.41	35.86

V. EXPERIMENTS

In order to analyze our method, we conducted comprehensive experiments on the synthetic data, EuRoc dataset, and TUM VI benchmark dataset. Besides, we also provided an intuitive demonstration: https://youtu.be/OnMAsmDsTVE, the related code of our algorithm is released at https://github.com/xubogithub/Structural-and-Non-structural-line. In these experiments, the algorithm ran on a computer with Intel Core i7-9750H@ 2.6 GHz, 16 GB memory and ROS Kinetic [24], ceres 2.0.0 [25].

To assess the advantages of our proposed approach, we compared our method with VINS-Mono without loop closure [2], PL-VIO [7], PL-VINS without loop closure [26], openVINS with monocular mode [27], and BASALT [28]. We choose the absolute pose error (APE) as the main evaluation metric which directly compares the trajectory error between the estimated pose and the groundtruth. The open-source accuracy evaluation tool evo [29] was used to evaluate the trajectory accuracy.

A. Synthetic Data

To verify the validity of non-structural lines' parametric expression, we generated a synthetic environment in which all 3D lines make up a room as groundtruth. As is shown in Fig. 5, the green lines in (a) are the simulated 80 landmarks, the blue track is composed of 600 VIO poses, each pose forms line observations in the camera frame, red lines in (b) are reconstructed by our method while there is no noise added to the line observations. In the experiment, we optimized all 3D line landmarks with our 2-parameter expression, the 2-parameter expression in [30], and the 4-parameter expression in [7] respectively. The reconstruction accuracy is obtained by comparing the end-points of reconstructed straight lines with the simulated 3D line landmarks. As shown in Table I, our 2-parameter expression has comparable accuracy compared with the 4-parameter expression, but the calculation time in half and the efficiency is greatly improved,

TABLE II
THE RMSE OF THE DIFFERENT METHODS ON EuRoc DATASET. THE TRANSLATION (CM) AND ROTATION (DEG) ERROR ARE LIST AS FOLLOWS. IN **BOLD** THE BEST RESULT

sequence		MH_01	MH_02	MH_03	MH_04	MH_05	V1_01	V1_02	V1_03	V2_01	V2_02	V2_03
VINS-Mono	trans.	16.8	17.1	19.4	34.6	29.2	**8.7**	7.9	20.7	8.2	15.7	20.4
	rot.	1.4	2.3	1.6	1.5	**0.7**	6.3	2.6	6.2	2.0	4.3	2.7
PL-VIO, 4-parameter	trans.	15.1	12.9	14.0	27.2	24.4	9.7	7.9	13.9	8.7	10.4	13.5
	rot.	2.9	2.1	1.3	2.5	1.1	6.1	1.8	3.7	2.5	2.1	2.6
PL-VIO, 2-parameter	trans.	15.5	13.5	14.4	30.3	23.5	10.5	8.9	14.1	8.3	11.3	14.3
	rot.	3.0	2.5	1.3	2.5	1.3	6.1	1.8	3.9	2.5	1.9	2.5
PLS-VIO-LBD	trans.	14.0	10.9	12.5	14.1	19.4	10.6	7.8	12.8	9.5	11.4	12.8
	rot.	2.8	1.5	**1.2**	0.9	1.0	**6.0**	1.5	3.0	1.0	2.1	2.8
PLS-VIO	trans.	**12.8**	**9.3**	**10.6**	**13.4**	16.8	9.8	**7.3**	**10.8**	**7.7**	10.6	**11.0**
	rot.	**1.3**	0.9	**1.2**	0.9	**0.7**	**6.0**	**1.3**	2.3	**0.6**	**1.5**	**1.5**
PS-VIO	trans.	12.9	9.4	10.9	13.6	**16.0**	10.3	8.5	11.9	9.7	**10.3**	13.1
	rot.	1.4	**0.6**	1.4	**0.8**	**0.7**	6.1	1.7	**2.1**	1.0	1.9	1.7

the method in [30] optimizes two coefficients corresponding to the orthonormal bases of the line's direction vector, our method has the same accuracy and time consuming compared with it, which shows that the 2-parameter expression of the line can improve the efficiency of line optimization.

B. EuRoc Dataset

The EuRoc micro aerial vehicle (MAV) datasets consist of two scenes, a machine hall and an ordinary room, which contain structure and non-structure scenes, In our experiments, we used the images from the left camera, besides, the extrinsic and intrinsic parameters are specified by the dataset.

1) Localization Accuracy: To illustrate the effect of the introduced structural line constraints and non-structural line constraints in the VIO system, the behavior of different parameter expressions of non-structural line features, as well as the superiority of our line matching method compared with the LBD method [31], we conducted ablation experiments to evaluate the RMSE APE of multiple systems on the EuRoc datasets. Methods used for comparison include: VINS-Mono [2] is only with point features, PL-VIO [7] combines points and non-structural line features, where LBD descriptors are used to match lines, the 4-parameter [7] and our 2-parameter representation of straight line are implemented in PL-VIO to compare the performance of different parameters, PLS-VIO-LBD has introduced structural constraints into the PL-VIO with 2-parameter representation of line, PLS-VIO replaced the LBD descriptor in the PLS-VIO-LBD system with our line matching algorithm, and PS-VIO removed the non-structural line features in the PLS-VIO. As shown in Table II, PLS-VIO achieves the smallest translation error on almost all the sequences, which means using both structural lines and non-structural lines in the VIO system can improve the accuracy and robustness of pose estimation, meanwhile, it also verifies the effectiveness of our line matching method. Specially, by comparing the experimental results of PLS-VIO-LBD with PL-VIO-2-parameter, we can find that introducing structural constraints into PL-VIO can significantly improve the accuracy of the system when there are enough structural straight lines in the scene, such as MH_04 and MH_05 sequences. At the same time, from the experiment results of PLS-VIO and PS-VIO, we
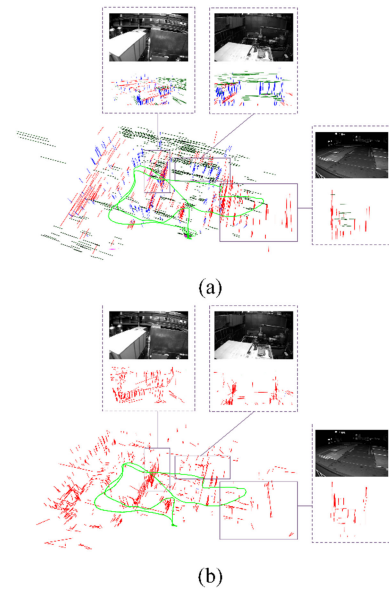

(a)

(b)

Fig. 6. Line map of EuRoc MH_05 dataset generated by (a) PLS-VIO, red, green and blue lines in the map are landmarks of structrual line features in X, Y, Z direction, the purple lines are the landmarks of non-structural line features, (b) PL-VIO, red lines in the map are landmarks of line features.

can see that using both structural lines and non-structural lines achieves better results than only using structural lines. PLS-VIO has achieved a lower RMSE, especially on the V series, the error was reduced by an average of 11%. The environment of V sequences is messy and there are many non-structural lines, the constraints of which can play a better role.

Meanwhile, the expression of 2-paramater of lines is easily affected by the observation noise of straight line, the positioning accuracy of PL-VIO with 2-parameter is lower than that of 4-parameter which sacrifices accuracy to maintain efficiency. By comparing PLS-VIO and PLS-VIO-LBD, our matching method can find accurate matching lines in complex environments, thereby improving the estimation of pose.

2) Mapping Quality: We compared the line map constructed by PLS-VIO proposed in this paper with that constructed by PL-VIO to evaluate the map quality. Fig. 6(a) shows a bird eye view of the map for PLS-VIO, as well as three detailed sub-maps
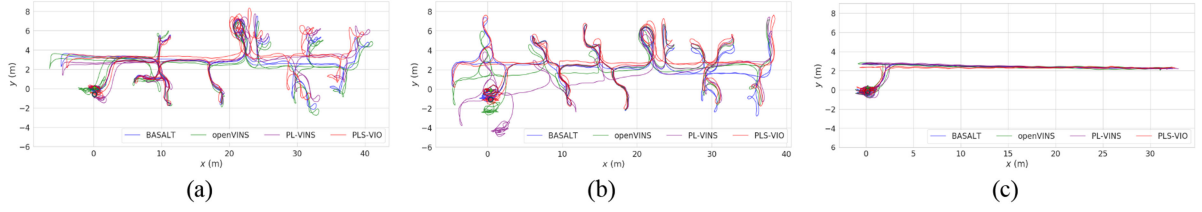
Fig. 7. The trajectories of BASALT, openVINS, PL-VINS and PLS-VIO running on (a) Corridor1 dataset, (b) Corridor2 dataset, (c) Corridor4 dataset. Cross is the start of trajectory, and circle is the end of trajectory. The corridor is along the horizontal direction, we aligned the four trajectories at the initial time with the groundtruth, the end position of PLS-VIO is close to the groundtruth and the drift of orientation is small.

TABLE III
THE RMSE OF THE STATE-OF-ART METHODS COMPARE TO OUR PLS-VIO ON TUM VI BENCHMARK DATASET. THE TRANSLATION (CM) ERRORS ARE LIST AS FOLLOWS. IN **BOLD** THE BEST RESULT

Seq.	BASALT	openVINS	PL-VINS	PLS-VIO	Length (m)
corridor1	23.2	63.2	32.5	**18.6**	305
corridor2	45.8	46.4	70.0	**33.7**	322
corridor3	39.1	35.5	67.8	**26.1**	300
corridor4	22.1	33.5	24.1	**21.4**	114
corridor5	44.9	**32.1**	49.4	37.6	270

TABLE IV
MEAN EXECUTION TIME (UNIT: MILLISECOND) OF PL-VIO WITH 2-PARAMETER, WITH 4-PARAMETER, PS-VIO AND PLS-VIO RUNNING ON THE V2_02 SEQUENCE

Module	PL-VIO 2-parameter	PL-VIO 4-parameter	PS-VIO	PLS-VIO
Line Extract	15.35	15.29	15.21	15.32
Line Match	6.66	6.52	6.49	6.79
Line Classify	0.00	0.00	0.08	0.08
Optimization	11.78	12.85	12.05	12.69
Marginalization	7.08	8.25	9.51	10.33

and their corresponding real scene images. The line landmarks in the map can well reflect the line features of the man-made machine hall. In order to better reflect the map quality of our system, we also run PL-VIO to build the line map, as shown in Fig. 6(b), we can observe that the map of PLS-VIO has more line features and the position of lines is also more accurate, at the same time, the accuracy improvement of the local map constructed by the VIO system will also affect the estimated pose, thereby improve the accuracy of the pose estimation.

C. TUM VI Benchmark Dataset

We tested the algorithm on corridor sequences of the TUM VI benchmark dataset, the corridor scene has a typical Manhattan man-made structure and contains obvious illumination changes and weak texture, which is great challenge to the algorithms. We ran BASALT, openVINS, PL-VINS, and PLS-VIO on corridor sequences, and drew the trajectories of four systems, openVINS and BASALT only use the point features and track points with optical flow method, which are affected by the illumination changes and weak texture environment, PL-VINS leverages both of the point and non-structural line features, however, due to the lack of global constraints of Manhattan regularity, the drift of orientation is large, as is shown in Fig. 7. From Table III, we can observe that our system achieves the smallest translation error on almost all the sequences, meanwhile, in the map of PLS-VIO, the landmarks of structural line feature in X, Y, Z direction are parallel to the main direction of the Manhattan world, which is shown in Fig. 1, the experimental results show that our method can effectively improve the accuracy of pose estimation and mapping quality.

D. Runtime Evaluation

We added structural lines and non-structural lines into the VIO based on point features, the time consuming is increased when we extract, match and classify lines as well as optimize them in the state estimator, we compared the runtime of PL-VIO with 2-parameter line, PL-VIO with 4-parameter line, PS-VIO, and PLS-VIO, here we mainly compared the average execution time of the frontend module about line features and backend module, which are evaluated on the V2_02 sequence. From Table IV, we can observe that compared with the LBD matching method, we used the 2D-2D and 2D-3D matching methods to improve the matching accuracy, but the time consumption is basically the same as that of LBD. For PL-VIO with 2-parameter and 4-parameter, the 2-parameter expression of line in this paper adds fewer state variables into the optimizer, therefore, the optimization and marginalization take less time. We added the non-structural lines into PS-VIO, which leads to the increase of matching time, optimization time and marginalization time, we counted the total runtime of PLS-VIO, which runs at 26.12 ms in the frontend, 33.09 ms in the backend, the system can fulfill the needs of running in real-time.

VI. CONCLUSION

In this paper, we present a novel VIO system fully exploiting the point, non-structural line and structural line features, which is called PLS-VIO. Both structural lines and non-structural lines are used to improve the accuracy and robustness of mapping and pose estimation, which is superior to the single type of lines. The 2-parameter representation of lines can speed up the optimization of the line landmarks, meanwhile, the 2D-2D and 2D-3D line matching methods reduce the mismatching of lines. The proposed state estimator is tested in large-scale scenes and corridor environments. The experiments show that the trajectory accuracy and mapping quality of our approach are better than the state-of-the-art visual-inertial odometry.

In the future, we will obtain structural information in the system by deep learning and improve the stability of structural line detection. Also, the planar constraints will be introduced for mapping.

REFERENCES

[1] 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," *IEEE Trans. Robot.*, vol. 37, no. 6, pp. 1874–1890, Dec. 2021.

[2] T. Qin, P. Li, and S. Shen, "VINS-Mono: A robust and versatile monocular visual-inertial state estimator," *IEEE Trans. Robot.*, vol. 34, no. 4, pp. 1004–1020, Aug. 2018.

[3] 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.

[4] S. Jianbo and C. Tomasi, "Good features to track," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, 1994, pp. 593–600.

[5] P. Smith, I. Reid, and A. Davison, "Real-time monocular SLAM with straight lines," in *Proc. Brit. Mach. Vis. Conf.*, Edinburgh, U.K., 2006, pp. 17–26.

[6] 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. Automat.*, 2017, pp. 4503–4508.

[7] H. Yijia, Z. Ji, G. Yue, H. Wenhao, and Y. Kui, "PL-VIO: Tightly-coupled monocular visual-inertial odometry using point and line features," *Sensors*, vol. 18, no. 4, 2018, Art. no. 1159.

[8] B. Xu, Y. Chen, S. Zhang, and J. Wang, "Improved point-line visual-inertial odometry system using helmert variance component estimation," *Remote Sens.*, vol. 12, no. 18, 2020, Art. no. 2901.

[9] D. G. Kottas and S. I. Roumeliotis, "Exploiting urban scenes for vision-aided inertial navigation," in *Proc. Robot.: Sci. Syst.*, 2013.

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

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

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

[13] J. M. Coughlan and A. L. Yuille, "Manhattan world: Compass direction from a single image by bayesian inference," in *Proc. 7th IEEE Int. Conf. Comput. Vis.*, 1999, vol. 2, pp. 941–947.

[14] 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.*, 2018, pp. 3686–3693.

[15] F. Camposeco and M. Pollefeys, "Using vanishing points to improve visual-inertial odometry," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2015, pp. 5219–5225.

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

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

[18] D. Schubert, T. Goll, N. Demmel, V. Usenko, J. Stückler, and D. Cremers, "The TUM VI benchmark for evaluating visual-inertial odometry," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2018, pp. 1680–1687.

[19] 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.

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

[21] B. Vincke, A. Elouardi, and A. Lambert, "Real time simultaneous localization and mapping: Towards low-cost multiprocessor embedded systems," *EURASIP J. Embedded Syst.*, vol. 2012, no. 1, pp. 1–14, 2012.

[22] L. Di Stefano, S. Mattoccia, and F. Tombari, "ZNCC-based template matching using bounded partial correlation," *Pattern Recognit. Lett.*, vol. 26, no. 14, pp. 2129–2134, 2005.

[23] 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.

[24] M. Quigley *et al.*, "ROS: An open-source robot operating system," in *Proc. ICRA Workshop Open Source Softw.*, vol. 3, no. 3.2. Kobe, Japan, vol. 3, no. 3.2, 2009, p. 5.

[25] S. Agarwal, K. Mierle, and Others, "Ceres solver," [Online]. Available: http://ceres-solver.org

[26] Q. Fu, J. Wang, H. Yu, I. Ali, F. Guo, and H. Zhang, "PL-VINS: Real-time monocular visual-inertial SLAM with point and line," 2020, *arXiv:2009.07462*.

[27] P. Geneva, K. Eckenhoff, W. Lee, Y. Yang, and G. Huang, "Openvins: A research platform for visual-inertial estimation," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2020, pp. 4666–4672.

[28] V. Usenko, N. Demmel, D. Schubert, J. Stückler, and D. Cremers, "Visual-inertial mapping with non-linear factor recovery," *IEEE Robot. Automat. Lett.*, vol. 5, no. 2, pp. 422–429, Apr. 2020.

[29] M. Grupp, "evo: Python package for the evaluation of odometry and SLAM," 2017. [Online]. Available: https://github.com/MichaelGrupp/evo

[30] J. Sola, T. Vidal-Calleja, and M. Devy, "Undelayed initialization of line segments in monocular SLAM," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst*, 2009, pp. 1553–1558.

[31] L. Zhang and R. Koch, "An efficient and robust line segment matching approach based on LBD descriptor and pairwise geometric consistency," *J. Vis. Commun. Image Representation*, vol. 24, no. 7, pp. 794–805, 2013.