Leveraging Structural Information to Improve Point Line Visual-Inertial Odometry

Bo Xu[1*], Peng Wang[2*], Yijia He[3], Yu Chen[1], Yongnan Chen[2], Ming Zhou[2]

***Abstract*— Leveraging line features can help to improve the localization accuracy of point-based monocular Visual-Inertial Odometry (VIO) system, as lines provide additional constraints. Moreover, in an artificial environment, some straight lines are parallel to each other. In this paper, we designed a VIO system based on points and straight lines, which divides straight lines into structural straight lines (that is, straight lines parallel to each other) and non-structural straight lines. In addition, unlike the orthogonal representation using four parameters to represent the 3D straight line, we only used two parameters to minimize the representation of the structural straight line and the non-structural straight line. Furthermore, we designed a straight line matching strategy based on sampling points to improve the efficiency and success rate of straight line matching. The effectiveness of our method is verified on both public datasets of EuRoc and TUM VI benchmark and compared with other state-of-the-art algorithms.**

I. INTRODUCTION

Simultaneous motion estimating and mapping are widely used in the field of intelligent robot, 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 egopose and build sparse map of 3D points [1]–[3]. However, in some textureless or illumination challenging environments, point-based VIOs can make the pose estimation fail [4].

Line features exist widespreadly in man-made environment, 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 the convergence speed is slow since non-structural lines have no effective directional constraints as structural lines do. Structural lines can be easily found in man-made environments, which can be abstracted as a set of blocks sharing three common dominant directions, known as Manhattan world [13]. Structural lines' directions depend on the corresponding vanishing point [14], which can provide effective geometric constraints to improve the accuracy of the SLAM system.

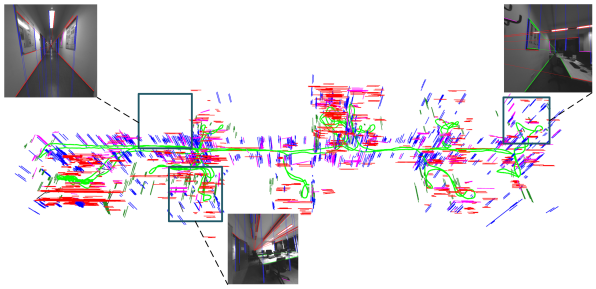

Fig. 1. The proposed monocular VIO system that builds line map. Red, green and blue lines in the map are the landmarks of structrual line feature in X, Y, Z direction, the purple lines are the landmarks of non-structural line feature. Three sub-images show the lines detected in the image.

In order to overcome the shortcomings of insufficient visual information in point-based VIO system, the line features are 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. [15] 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 error constraint.

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. [16] 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. [17] 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, both non-structural lines and structural lines have the potential to improve the robustness and accuracy of VIO system with different advantages. 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 features and structural line features to achieve accurate pose estimation

[1]Bo Xu, Yu Chen are with School of Geodesy and Geomatics, Wuhan University, Wuhan 430079, China; Corresponding author: Bo Xu, Email: boxu1995@whu.edu.cn

[2]Peng Wang, Yongnan Chen and Ming Zhou are with Faculty of Robot Science and Engineering, Northeastern University, Shenyang 110819, China

[3]Yijia He is with Kuaishou Technology, Beijing 100000, China

* Equal contribution

and point-line map construction, as shown in Fig.1. In order to reduce the calculation consumption in the process of line optimization. We adopt different parameterized expressions for non-structural line features. Furthermore, we describe the classification, matching and initialization of all the lines in detail. The main contributions of this work include:

- In order to make better use of the parallel structure information of lines that exits in the environment, we designed a line classification strategy to divide lines into non-structural lines and structural lines. The introduction of structural information can significantly improve the accuracy and efficiency of point-line VIO.
- In order to efficiently match and represent line features, we designed 2D-2D and 2D-3D line matching algorithms based on point feature tracking to improve the speed and accuracy of matching module. In addition, unlike the 4 parameter orthogonal expression of line, we introduced a 2-parameter representation method to improve the efficiency of line estimation.
- We provided a series of experiments to verify the effectiveness of the proposed algorithm. We run the system on the public datasets EuRoc [18] and TUM VI benchmark [19]. Experimental results demonstrated 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 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.

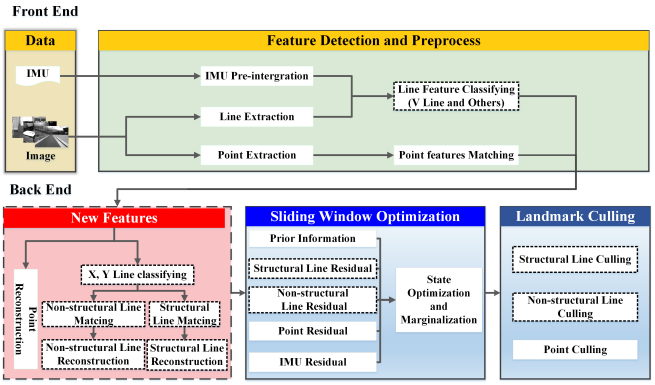

Fig. 2. Overview of our PLS-VIO system. The Front-End module is used to extract information from the raw measurement; Line classification and state variables are estimated with sliding window optimization in the Back-End.

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 are introduced. Second, we introduce the classification of the lines. Finally, the different matching method 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 $\mathbf{z} = \begin{bmatrix} u, v, 1 \end{bmatrix}^{\mathrm{T}}$ 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 expressed by $\mathcal{L} = \begin{bmatrix} {}^c\mathbf{n}^{\mathrm{T}}, {}^c\mathbf{v}^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}}$, 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 3D line $\mathcal{L}$ is on the plane π, we can use the local coordinate system $\{P\}$ on the plane π to represent $\mathcal{L}$. 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 amount 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: When a new structural line is detected, we use the representation in paper [17] to

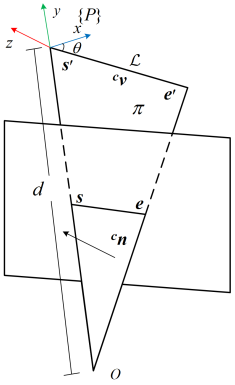

Fig. 3. the endpoint $\mathbf{s}'$ and $\mathbf{e}'$ of 3D line $\mathcal{L}$ are projected to the image plane to form $\mathbf{s}$ and $\mathbf{e}$. $\mathbf{s}'$, $\mathbf{e}'$ and optical center O of camera frame construct the plane π, the direction vector of $\mathcal{L}$ is ${}^{c}\mathbf{v}$, the 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.

parameterize this line. 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 agian 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 $\mathbf{l}_p^l = \begin{bmatrix} a, b, 0 \end{bmatrix}^{\mathrm{T}}$ on the XY plane, we use the inverse depth to represent the intersection point , namely $\begin{bmatrix} \theta, \rho, 0 \end{bmatrix}^{\mathrm{T}}$, where $\rho = \sqrt{a^2 + b^2}$ and $\theta = atan2(b, a)$, we adopt this representation to speed up the convergence of structural line.

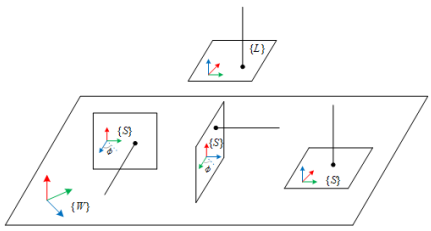

Fig. 4. Start frames $\{S\}$ of X, Y, Z direction structural lines from the parameter space $\{L\}$, the origin of start frame is in the world coordinate system $\{W\}$, the orientation of start frame is rotated about ϕ from the world coordinate system.

B. Structural Line and Non-structural Line Classification

We use the LSD algorithm [20] to detect lines in the image and then classify the structural lines from these lines, and the remaining unclassified lines are treated as non-structural lines. We use vanishing points in the image to recognize structural lines. For example, for classifying a structural line 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.

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 matched lines.

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 epipolar searching method [21] to find corresponding candidate matching points in the current frame. The ZMSSD (Zero-mean Sum of Squared Differences) template [22] 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 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 [23]. 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 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 current frame, the line's end points 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 affects 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 $\mathbf{s}^{c_1} = \begin{bmatrix} u_s, v_s, 1 \end{bmatrix}^{\mathrm{T}}$ and $\mathbf{e}^{c_1} = \begin{bmatrix} u_e, v_e, 1 \end{bmatrix}^{\mathrm{T}}$. Three

non-collinear points, including two endpoints of a line segment and the optical center O of camera, determine a plane $\boldsymbol{\pi} = \left[\pi_x, \pi_y, \pi_z, \pi_w\right]^{\mathrm{T}}$, 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:

$$ {}^{c}\mathcal{L}^{*} = \begin{bmatrix} [\mathbf{v}^{c}]_{\times} & \mathbf{n}^{c} \\ -\boldsymbol{n}^{c\mathrm{T}} & 0 \end{bmatrix} = \boldsymbol{\pi}_1\boldsymbol{\pi}_2^{\mathrm{T}} - \boldsymbol{\pi}_2\boldsymbol{\pi}_1^{\mathrm{T}} \in \mathbb{R}^{4\times4} \tag{1} $$

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

We get the Plücker coordinate of the line ${}^{c}\mathcal{L} = \left[{}^{c}\mathbf{n}^{\mathrm{T}}, {}^{c}\mathbf{v}^{\mathrm{T}}\right]^{\mathrm{T}}$ 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 ${}^{c}\mathcal{L} = \left[{}^{c}\mathbf{n}^{\mathrm{T}}, {}^{c}\mathbf{v}^{\mathrm{T}}\right]^{\mathrm{T}}$. And then we get the three-dimensional endpoints expression of line in the world coordinate ${}^{w}\mathcal{L} = \left[\mathbf{s}^{w\mathrm{T}}, \mathbf{e}^{w\mathrm{T}}\right]^{\mathrm{T}}$ using line trimming [24]. 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):

$$ {}^{w}\mathbf{p} = \begin{bmatrix} \mathbf{R}_S^W\mathbf{R}_L^S & \mathbf{P}_S^W \\ \mathbf{0} & 1 \end{bmatrix}^{-\mathrm{T}} {}^{l}\mathbf{p} \tag{2} $$

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.

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

We use the intersection ${}^{l}\mathbf{l}_p = \left[{}^{l}l_{px}, {}^{l}l_{py}, 0\right]^{\mathrm{T}}$ 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:

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

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 k^{th} 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:

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

where $\left[x^{c_j}, y^{c_j}, z^{c_j}\right]^{\mathrm{T}}$ is the projected point from the first observation frame of the feature. The covariance matrix $\sum_{\mathcal{F}}$ of point is based on the assuming that the standard deviation of the point features noise is 1.5 pixel.

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 ${}^{d}\mathbf{v} = \left[cos\theta, sin\theta, 0\right]^{\mathrm{T}}$, we can transfer the ${}^{d}\mathbf{v}$ from the local coordinate to the camera coordinate system to get the line direction ${}^{c}\mathbf{v}$.

$$ {}^{c}\mathbf{v} = \mathbf{R}_P^C {}^{d}\mathbf{v} \tag{6} $$

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

$$ \mathbf{s} = \frac{\mathbf{y}}{\|\mathbf{y}\|} \cdot d \tag{7} $$

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:

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

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

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

$$\mathbf{l}_{l}^{m_i} = [\mathbf{s}']_{\times}{}^{c}\mathbf{v}' \tag{10}$$

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

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

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

$$d(\mathbf{s}, \mathbf{l}) = \frac{\mathbf{s}^{\mathrm{T}}\mathbf{l}}{\sqrt{l_1^2 + l_2^2}} \tag{12}$$

For the setting of the non-structural line covariance $\sum_{\mathcal{L}}$, we assume that the noise is the same as point feature.

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 (13), 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.

$${}^{c}\mathbf{l}_p = \mathbf{R}_{W}^{C}\mathbf{R}_{S}^{W}\mathbf{R}_{L}^{S}{}^{l}\mathbf{l}_p + (\mathbf{R}_{W}^{C}\mathbf{P}_{S}^{W} + \mathbf{P}_{W}^{C}) \tag{13}$$

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. ${}^{l}\mathbf{l}_p = \begin{bmatrix} a, b, 0 \end{bmatrix}^{\mathrm{T}}$, 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 ${}^{l}\mathbf{v} = \begin{bmatrix} 0, 0, 1 \end{bmatrix}^{\mathrm{T}}$, and we use (14) to transform it into the camera frame coordinate.

$${}^{c}\mathbf{v} = \mathbf{R}_{W}^{C}\mathbf{R}_{S}^{W}\mathbf{R}_{L}^{S}{}^{l}\mathbf{v} \tag{14}$$

We get the line equation on the target frame by:

$$\mathbf{l}_{s}^{m_i} = [{}^{c}\mathbf{l}_p]_{\times}{}^{c}\mathbf{v} \tag{15}$$

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, because of the strong constraint information it carries, the standard deviation of the structural line is set to 0.15 pixels.

V. Experiments

In order to analyze our method, we conducted comprehensive experiments on the public EuRoC and TUM VI benchmark datasets. Besides, we also provided the code of our algorithm https://github.com/xubogithub/Structural-and-Non-structural-line. In these experiments, the algorithm ran on a computer with Intel Core i7-9750H@ 2.6GHz, 8GB memory and ROS Kinetic [25], ceres 2.0.0 [26].

To assess the advantages of our proposed approach, we compared our method with OKVIS with monocular mode [3], VINS-Mono without loop closure [2] and PL-VIO [7]. 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 [27] 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 that there is no noise added to the line observations. We also compared the consuming time and accuracy of 4-parameter orthogonal representation and 2-parameter representation of line proposed in this paper as the line observation noise increases, as shown in Tab. I, Tab.II, in the experiment, we optimized the whole scene structure adding 80 line landmarks and 600 poses to the optimizer at one time, the accuracy is obtained by comparing the reconstructed straight lines with the simulated 3D line landmarks. From the experimental results, we can see that our 2-parameter expression and 4-parameter expression have the same optimization accuracy on the whole, but we optimized fewer parameters, the running time is only half of the 4-parameter expression, which improves the efficiency.

TABLE I
Comparison of consuming time[s] of different line representation in different pixel noise level[pixel]

noise [pixels]	0	1,5	3	4.5
2-parameter	20.54	20.34	20.38	20.66
4-parameter	40.89	41.00	40.75	40.49

TABLE II

Comparison of accuracy[cm] of different line representation in different pixel noise level[pixel]

noise [pixels]	0	1,5	3	4.5
2-parameter	0.06	7.93	14.99	20.63
4-parameter	0.05	5.34	10.65	16.17

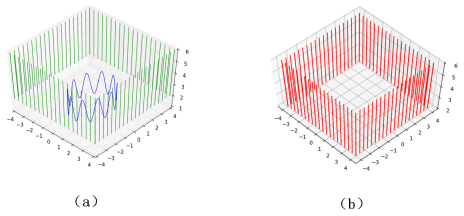

Fig. 5. Synthetic data and reconstruction of 3D lines by our method. (a) groundtruth. (b) our method.

TABLE III

The RMSE of the state-of-art methods compare to our PLS-VIO on EuRoc dataset. The translation (cm) and rotation (deg) error are list as follows. In **bold** the best result

	OKVIS		VINS		PL-VIO		PLS-VIO	
Seq.	trans.	rot.	trans.	rot.	trans.	rot.	trans.	rot.
MH_01	29.5	3.2	14.8	2.0	20.1	**1.6**	**11.1**	**1.6**
MH_02	30.7	3.9	17.1	2.3	13.1	1.7	**9.3**	**0.9**
MH_03	33.4	3.3	19.4	1.6	26.1	1.7	**15.7**	**0.8**
MH_04	38.9	2.3	34.6	1.5	35.8	1.6	**17.1**	**1.4**
MH_05	46.7	2.4	29.2	**0.7**	24.4	1.1	**14.4**	**0.7**
V1_02	22.2	6.0	**7.9**	2.6	17.0	3.2	8.9	**1.5**
V1_03	28.1	8.1	20.7	6.2	27.0	**3.4**	**14.3**	4.2
V2_01	14.0	2.2	8.2	**2.0**	9.3	2.2	**7.4**	2.3
V2_02	21.1	4.9	15.7	4.3	12.3	2.9	**12.2**	**1.7**

B. EuRoc Dataset

The EuRoc micro aerial vehicle(MAV) datasets consist of two scenes, a machine hall and an ordinary room, the datasets contain stereo images from a global shutter camera at 20FPS and synchronized IMU measurements at 200 Hz. The datasets provide all the extrinsic, intrinsic parameters and groundtruth trajectory. Besides, our system is a monocular-VIO, we only used the images from the left camera.

1) Localization Accuracy: we evaluated the RMSE APE of OKVIS, VINS, PL-VIO and PLS-VIO proposed in this paper on EuRoc datasets. Tab. III shows the result of different methods, our approach using points, non-structural lines and structural lines achieves the smallest translation error on all the sequences except for V1_02. In addition, our approach achieves the smallest rotation error on most of the sequences except for V1_03 and V2_01. The reason is that the MH sequences have much more structural lines than the non-structural lines, which could offer better constraint to the system. On the contrary, the V sequences have more non-structural lines, the constraint information may not be so strong. According to the result of pose estimation, the system proposed in this paper can effectively decrease the accumulation error of translation and rotation when running in the man-made buildings. Basically, our system with structural lines and non-structural lines can improve the robustness of estimator in the complex scenes.

Further analysis of the results on MH_05, as shown in Fig.6, the translation error of PLS-VIO with point, non-structural line and structural line features is smallest, and PL-VIO with point and non-structural line features presents lower translation error in 20~40s and after 80 seconds compared with VINS with point features. It can be seen that the VIO using line features can improve the accuracy of pose estimation, especially structural line features. We also compared the success rate of the LBD matching method [28] used in PL-VIO and the line matching method prpposed in this paper on MH_05 dataset, the method to calculate the success matching rate is: after obtaining the matching lines by the algorithm, the groundtruth of dataset is used to project the mathcing lines in the previous frame into the current frame to calculate the projection error. We count the line whose projection error is less than a certain threshold and calculate the success matching rate. As shown in Fig.7, the success matching rate of proposed line matching method(red curve) is higher than that of LBD(blue curve), the average success matching rates of two algorithms are 0.49 vs 0.38. We also compared the matching time of each frame, the matching time of algorithm proposed is shorter than that of LBD, the average time consuming of our algorithm is 6.3ms and LBD is 8.0ms. By improving the robustness of line matching method, we can improve the accuracy of pose estimation and corresponding mapping quality.

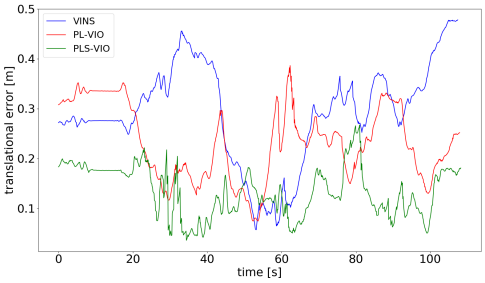

Fig. 6. Translation error of APE on the dataset MH_05 with VINS, PL-VIO, PLS-VIO

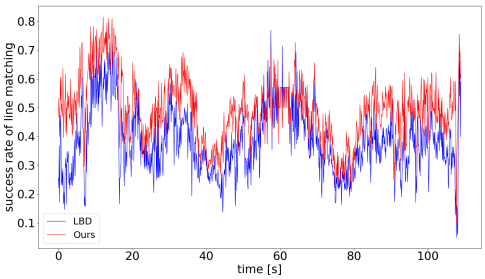

Fig. 7. Comparison of success rate between lbd line matching method and the method proposed in this paper on MH_05 dataset

2) Mapping Quality: We compared the line map constructed by PLS-VIO proposed in this paper with the line map constructed by PL-VIO to evaluate the map quality. Fig.8 shows a bird eye view of the map for PLS-VIO, as well as three detailed sub-maps and their corresponding real scene images. Adopting the parameter representation of lines used in this paper, the line landmarks converge more quickly. The line landmarks in the map can well reflect the structural 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.9, both the number of lines and the accuracy of line landmarks in the map have been greatly improved. The accuracy improvement of the local map constructed by the VIO system will also impose stronger constraints on the estimated pose, thereby improve the accuracy of the pose estimation.

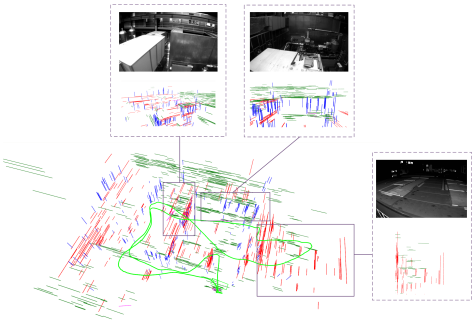

Fig. 8. Line map of EuRoc MH_05 dataset generated by PLS-VIO. Red, green and blue lines in the map are landmarks of structrual line features in X, Y, Z direction, the purple lines are the landmarks of non-structural line features. Three sub-images show the detail of the maps and real scene images.

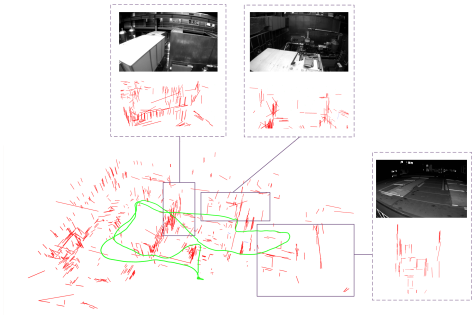

Fig. 9. Line map of EuRoc MH_05 dataset generated by PL-VIO. Red lines in the map are landmarks of line features, three sub-images show the detail of the maps and real scene images

C. TUM VI Benchmark dataset

We tested the algorithm on corridor1 of the TUM VI benchmark datasets, the scene of corridor1 dataset contains a long corridor and several office rooms, the dataset provides camera image with 512×512 at 20 Hz as well as IMU measurements of accelerations and angular velocities on 3 axes at 200 HZ, the dataset also provides all the extrinsic and intrinsic parameters and the accurate pose groundtruth at the start and end of the sequences. The corridor scene has a typical Manhattan man-made structure which is suitable for the construction of line map.

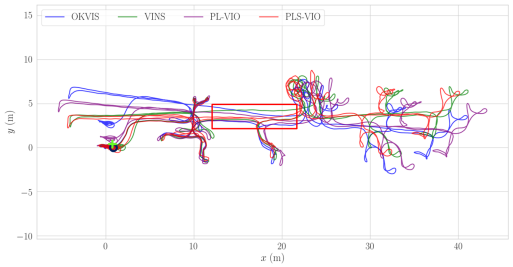

Fig. 10. The trajectories of OKVIS, VINS, PL-VIO and PLS-VIO running on corridor1 dataset. Cross is the start of trajectory, and circle is the end of trajectory, the red box indicates the trajectories when the collection device enters into the corridor after coming out of the office.

TABLE IV

The RMSE of the state-of-art methods compare to our PLS-VIO on TUM VI benchmark dataset. The translation (cm) error are list as follows. In **bold** the best result

system.	OKVIS	VINS	PL-VIO	PLS-VIO
trans	55.5	44.6	52.9	**31.7**

We ran OKVIS, VINS, PL-VIO and PLS-VIO on corridor1, and drew the trajectories of four systems, as shown in Fig.10, the real trajectory is parallel along the corridor, aligning the start of the sequence, we can see that the PL-VIO and OKVIS have obvious drifts, when the collection device comes out of the room, as shown by the red box in the Fig.10, while the trajectory of VINS also drifts. We used RMSE APE to evaluate accuracy of the start-end trajectory and the groundtruth, as shown in Tab.IV, corridor1 is long corridor scene which mainly contains textureless walls and lack of sufficient illumination, the camera movement is also violent, so the quality of point feature is poor, using line segment as the supplement can effectively improve the accuracy of the pose estimation, therefore, the trajectory error of PLS-VIO with structural line and non-structural line features is the smallest. We also evaluated mapping quality of PLS-VIO and PL-VIO, which is shown in Fig.11 and Fig.12, respectively. In the map of PLS-VIO, red, green and blue lines are landmarks of structural line feature in X, Y, Z direction, which are parallel to the main direction of the Manhattan world, the purple lines are the landmarks of non-structural line feature, which are less than the structural lines. As visual information supplement to VIO with point features, using the structural lines and non-structural lines can effectively improve the accuracy of pose estimation and mapping quality.

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. To speed up the convergence of the

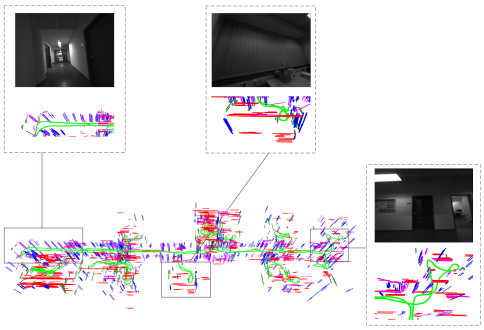

Fig. 11. Line map of TUM corridor1 dataset generated by PLS-VIO. Red, green and blue lines in the map are landmarks of structrual line feature in X, Y, Z direction, the purple lines are the landmarks of non-structural line feature. Three sub-images show the detail of the map and real scene image.

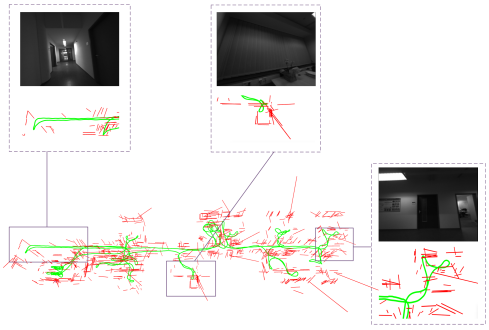

Fig. 12. Line map of TUM corridor1 dataset generated by PL-VIO. The red lines in the map are landmarks of line feature, three sub-images show the detail of the map and real scene image.

line landmarks, we adopt two different line parameter representations for non-structural line and structural line features. For further improve the robustness of the line landmarks in the VIO system, we utilize two different initialization methods of line, and employ the frame-to-frame and frame-to-map line matching methods. The proposed state estimator is tested in large-scale scene and corridor environment. 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.

ACKNOWLEDGMENT

This work was supported by the National key R&D Program of china (Grant No. 2020YFD1100200).

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 multi-map slam," *arXiv preprint arXiv:2007.11898*, 2020.

[2] T. Qin, P. Li, and S. Shen, "Vins-mono: A robust and versatile monocular visual-inertial state estimator," *IEEE Transactions on Robotics*, vol. 34, no. 4, pp. 1004–1020, 2018.

[3] S. Leutenegger, S. Lynen, M. Bosse, R. Siegwart, and P. Furgale, "Keyframe-based visual – inertial odometry using nonlinear optimization," 2014.

[4] S. Jianbo and C. Tomasi, "Good features to track," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1994, pp. 593–600.

[5] P. Smith, I. Reid, and A. Davison, "Real-Time Monocular SLAM with Straight Lines."

[6] A. Pumarola, A. Vakhitov, A. Agudo, and A. Sanfeliu, "PL-SLAM : Real-Time Monocular Visual SLAM with Points and Lines," pp. 4503–4508, 2017.

[7] 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.

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

[9] D. G. Kottas and S. I. Roumeliotis, "Exploiting Urban Scenes for Vision-aided Inertial Navigation." in *Robotics: Science and Systems*, 2013.

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

[11] G. Zhang, D. H. Kang, I. H. Suh, and S. Member, "Loop Closure Through Vanishing Points in a Line-based Monocular SLAM," pp. 4565–4570, 2012.

[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," pp. 936–941, 2009.

[13] J. M. Coughlan and A. L. Yuille, "Manhattan world: Compass direction from a single image by bayesian inference," in *Proceedings of the seventh IEEE international conference on computer vision*, vol. 2. IEEE, 1999, pp. 941–947.

[14] 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 *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, 2017, pp. 345–353.

[15] 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," pp. 3686–3693, 2018.

[16] F. Camposeco and M. Pollefeys, "Using vanishing points to improve visual-inertial odometry," in *2015 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2015, pp. 5219–5225.

[17] D. Zou, Y. Wu, S. Member, L. Pei, H. Ling, and W. Yu, "StructVIO : Visual-Inertial Odometry With Structural Regularity of Man-Made Environments," vol. 35, no. 4, pp. 999–1013, 2019.

[18] M. Burri, J. Nikolic, P. Gohl, T. Schneider, J. Rehder, S. Omari, M. W. Achtelik, and R. Siegwart, "The EuRoC micro aerial vehicle datasets," vol. 35, no. 10, pp. 1157–1163, 2016.

[19] D. Schubert, T. Goll, N. Demmel, V. Usenko, and D. Cremers, "The TUM VI Benchmark for Evaluating Visual-Inertial Odometry," pp. 1680–1687, 2018.

[20] J.-m. Morel, G. Randall, and R. G. V. Gioi, "LSD : A Fast Line Segment Detector with a False Detection Control," vol. 32, no. 4, pp. 722–732, 2010.

[21] J. Engel, V. Koltun, and D. Cremers, "Direct Sparse Odometry," vol. 8828, no. c, 2017.

[22] S. Weiss and J. T. Hinzmann, "Robust vision-based navigation for micro air vehicles."

[23] L. D. Stefano, S. Mattoccia, and F. Tombari, "ZNCC-based template matching using bounded partial correlation," vol. 26, pp. 2129–2134, 2005.

[24] G. Zhang, J. H. Lee, J. Lim, and I. H. Suh, "Building a 3-d line-based map using stereo slam," *IEEE Transactions on Robotics*, vol. 31, no. 6, pp. 1364–1377, 2015.

[25] M. Quigley, B. Gerkey, K. Conley, J. Faust, T. Foote, J. Leibs, E. Berger, R. Wheeler, and A. Ng, "ROS : an open-source Robot Operating System," no. Figure 1.

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

[27] M. Grupp, "evo: Python package for the evaluation of odometry and slam." https://github.com/MichaelGrupp/evo, 2017.

[28] L. Zhang and R. Koch, "An efficient and robust line segment matching approach based on lbd descriptor and pairwise geometric consistency," *Journal of Visual Communication and Image Representation*, vol. 24, no. 7, pp. 794–805, 2013.