Structural Regularity Aided Visual-Inertial Odometry With Novel Coordinate Alignment and Line Triangulation

Hao Wei, Fulin Tang, Zewen Xu, and Yihong Wu

Abstract—**Man-made buildings exhibit structural regularity, which can provide strongly geometrical constraints for Visual-Inertial Odometry (VIO) systems. To make full use of the structural information, we propose a new structural regularity aided VIO with novel coordinate alignment and line triangulation under Manhattan world assumption. The proposed VIO system is built upon OpenVINS [1] and partly based on our previous work [2]. The proposed coordinate alignment method makes the Jacobians and reprojection errors become concise but also makes the required computation number for transformations decrease. In addition, a novel structural line triangulation method is provided, in which the global orientation of a structural line is used to refine its 3D position. All the novelties result in a more accurate and fast VIO system. The system is tested on EuRoC MAV dataset and a self-collected dataset. Experimental results demonstrate that the proposed method obtains better accuracy compared with state-of-the-art (SOTA) point-based systems (VINS-Mono [3] and OpenVINS [1]), point-line-based systems (PL-VINS [4] and Wei et al. [2]), and structural line-based system (StructVIO [5]). Notably, the self-collected dataset is recorded on Manhattan world scenes, and is also full of challenging weak texture and motion blur situations. On the dataset, the accuracy of our method is increased by 40.7% compared with the SOTA point-line-based systems.**

Index Terms—**Visual-Inertial SLAM, SLAM.**

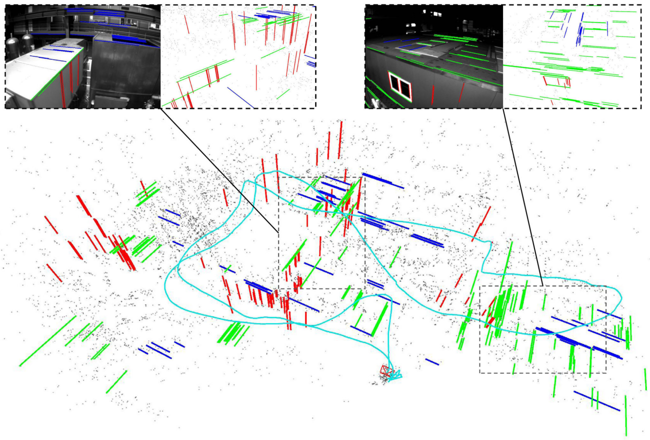


Fig. 1. Results of the proposed VIO system on Machine Hall 05 sequence of EuRoC MAV dataset, where the blue, green, and red lines represent 3D structural lines in X, Y, and Z directions, respectively, camera trajectory is marked as cyan and the small gray points are point feature landmarks. 2D structural lines are also shown in the top images.

I. Introduction

VISUAL-INERTIAL Odometry (VIO) and Simultaneous Localization and Mapping (SLAM) are fundamental technologies for various applications, such as unmanned aerial vehicle, virtual/augmented reality, and robot navigation [6]. Most existing SLAM/VIO systems rely on point features, which may degenerate or even collapse in some difficult scenes. In the field of SLAM/VIO, multiple types of image features are complementary to each other, and their combination leads to a more versatile and robust system [7]. Line is one of the most important image features, which is widely distributed in man-made environments. Therefore, combining point and line features to improve the accuracy and robustness of SLAM/VIO systems has been investigated in many studies [8]–[11].

Under Manhattan world assumption, there are two kinds of line features, namely structural lines and non-structural lines [12], [13]. As shown in Fig. 1, structural lines are lines that directions are consistent with the Manhattan world buildings' dominant directions. The global directions of structural lines can be obtained easily and can provide strong geometrical constraints, which are useful to improve the accuracy of VIO/SLAM [12].

However, the existing methods utilize structural lines in a complex and indirect way, which leads to complex forms for computing Jacobians and reprojection errors. To address this problem, we propose an efficient coordinate alignment method in this paper. Then, three axes of the global frame are aligned with a building's dominant directions. Following the alignment, structural lines are parameterized as 2D points on their corresponding projection planes, and then their Plücker coordinates are obtained without complex coordinate transformations, which leads to concise forms of Jacobians and reprojection errors.

Manuscript received 23 February 2022; accepted 11 July 2022. Date of publication 27 July 2022; date of current version 8 August 2022. This letter was recommended for publication by Associate Editor F. Fraundorfer and Editor S. Behnke upon evaluation of the reviewers' comments. This work was supported by the National Natural Science Foundation of China under Grants 61836015 and 62002359. *(Corresponding authors: Fulin Tang; Yihong Wu.)*

Hao Wei, Zewen Xu, and Yihong Wu are with the School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing 100190, China, and also with the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China (e-mail: weihao2019@ia.ac.cn; xuzewen2020@ia.ac.cn; yihong.wu@ia.ac.cn).

Fulin Tang is with the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China (e-mail: fulin.tang@nlpr.ia.ac.cn).

This letter has supplementary downloadable material available at https://doi.org/10.1109/LRA.2022.3194329, provided by the authors.

Digital Object Identifier 10.1109/LRA.2022.3194329

2377-3766 © 2022 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See https://www.ieee.org/publications/rights/index.html for more information.

Authorized licensed use limited to: INSTITUTE OF AUTOMATION CAS. Downloaded on September 05,2022 at 01:24:51 UTC from IEEE Xplore. Restrictions apply.

In addition, the accuracy of a VIO system is greatly affected by the precision of feature positions [14]. The existing structural line based methods use the traditional line triangulation algorithm [13] or the inverse depth representation to get the initial positions of structural lines [5]. However, the initial 3D positions exhibit a high uncertainty. For structural line features, their global directions are known after a line classification step. Therefore, we propose a novel structural line triangulation algorithm, in which global directions of the structural lines are used to refine their 3D positions after acquiring the initial positions.

The main contributions of this work are summarized as follows:

1) An efficient coordinate alignment method is presented for utilizing structural information, where the global frame is aligned with the dominant directions of buildings. The alignment makes the required transformation number decrease, which results in the computations of Jacobians and reprojection errors greatly easy, so that the proposed VIO system is more accurate, fast, and is of lower computation consumption.
2) A novel structural line triangulation algorithm is provided, in which global orientations of the structural lines are used to refine their 3D positions. More accurate line positions lead to a more accurate VIO system.
3) A VIO system by integrating the above coordinate alignment method and structural line triangulation algorithm is proposed, where we leverage point and structural line features. This system is built upon OpenVINS [1] and partly based on our previous work [2]. Specially, a given 2D projection parametrization and an accurate line classification make it much easier to process structural lines.
4) We made experiments by plugging our methods into OpenVINS. The results show that the proposed method achieves the best accuracy compared with SOTA point-based systems (VINS-Mono [3] and OpenVINS [1]), point-line-based systems (PL-VINS [4] and Wei et al. [2]), and structural line based system (StructVIO [5]), which is benefited from the proposed novelties.

II. Related Work

Compared with natural scenes, artificial buildings have strongly structural regularity, which can be used to improve the accuracy of SLAM/VIO. There are different strategies for leveraging structural information, which can be broadly categorized into direct methods and model-based methods.

Direct methods use particular regularities like parallelism, orthogonality, and coplanarity directly. For instance, Zhang et al. [15], [16] presented a line-based EKF-SLAM system that utilizes vertical and floor lines, and Lee et al. [17] proposed a visual-inertial SLAM system with parallel lines' constraints.

Model-based methods assume that the surrounding environments obey known models (such as Manhattan world [18] and Atlanta world [19]), and use the properties of the known distributions to improve SLAM/VIO systems. Some systems utilize structural regularity via vanishing points (VPs). A VP is the intersection of a group line features that are projected from parallel 3D lines [20], and there are three orthogonal VPs in a Manhattan scene. Camposeco et al. [21] added VPs in the state vector to remove the angular drift of VIO. Kim et al. [22] used VPs to obtain drift-free rotation. Besides, [11], [23] and [24] adopt the alike technique. Other methods use structural lines as image features. Kottas et al. [25] presented an extended Kalman filter (EKF) based VIO system for utilizing measurements of structural lines. Zou et al. [26] extended the standard EKF visual SLAM to adopt the building structural lines and points as features under Manhattan world assumption, which is named as StructSLAM. StructVIO [5] is a VIO system that adopts structural line features in an Atlanta world model, which are very accurate and robust in different kinds of complex environments. Xu et al. [13] used both structural and non-structural lines to improve the accuracy of an optimization-based VIO system.

III. System Overview

As shown in Fig. 2, the proposed VIO system includes four main procedures: initialization, image processing, propagation, and update. The initialization procedure in this paper includes the classical VIO initialization step [3] and coordinate alignment, which are presented in Section IV. Processing of structural lines is given in Section V and then structural line triangulation algorithm is given in Section VI.

IV. Coordinate Alignment

The goal of VIO initialization is to get good initial values for the inertial variables: body velocity, gravity direction, and IMU biases [27]. After the step, we can get the rotation ${}^{I_0}_{G}\mathbf{R}$ from the global frame $\{\mathrm{G}\}$ to the initial IMU frame $\{I_0\}$. In a Manhattan world building, the dominant directions provide a natural coordinate frame, which is termed as the building frame $\{\mathrm{B}\}$. For utilizing the structural lines easily, Manhattan world dominant directions are defined as the global coordinate ($\{\mathrm{G}\}$) axes X, Y, Z in this paper, lines whose directions are the same as the axis directions are defined as structural lines. We make the $\{\mathrm{G}\}$ aligned with the building frame $\{\mathrm{B}\}$ by computing the angle θ in Fig. 3 accurately.

After the VIO initialization [3], we use the algorithm in [28] to detect VPs and structural lines on the input images. Notably, directions of the detected VPs are not determined. Since Z axis is aligned with gravity, the method in Section V-B is also used to find the VP in Z direction. Meanwhile, we monitor the number of structural lines in X and Y directions to check whether the environment conforms to the Manhattan world assumption. If the number is greater than a preset threshold, the VPs in X and Y directions of the current image are utilized to calculate θ.

Different from [25], the system input of the proposed alignment method is VPs. Let $bfv_i(i = x, y, z)$ denote VPs detected on the current image, $\mathbf{K}$ be the intrinsic matrix of camera, $\mathbf{e}_x$, $\mathbf{e}_y$, and $\mathbf{e}_z$ be the direction units of X, Y and Z in $\{\mathrm{G}\}$, where:

$$\mathbf{e}_x = \begin{bmatrix}1\\0\\0\end{bmatrix}, \quad \mathbf{e}_y = \begin{bmatrix}0\\1\\0\end{bmatrix}, \quad \mathbf{e}_z = \begin{bmatrix}0\\0\\1\end{bmatrix}. \tag{1}$$

We distinguish the VP in X direction from two horizontal VPs by checking the following arccosine value φ_j between ${}^{G}\mathbf{e}_x$ and $\mathbf{v}_i$:

$$\varphi_j = arccos\frac{\left({}^{C}_{G}\mathbf{R}{}^{G}\mathbf{e}_x\right)^T\mathbf{K}^{-1}\mathbf{v}_i}{\|\mathbf{K}^{-1}\mathbf{v}_i\|}, \quad i = x,\, y, \quad j = 1, 2 \tag{2}$$

where ${}^{C}_{G}\mathbf{R}$ is the rotation from $\{\mathrm{G}\}$ to the current camera frame $\{\mathrm{C}\}$. The VP corresponding to the smaller one of the two values ($\varphi_x = \min(\varphi_1, \varphi_2)$) are determined as the VP in X direction. Then, we calculate φ_y similar to (2). To avoid confusion of θ, we

Authorized licensed use limited to: INSTITUTE OF AUTOMATION CAS. Downloaded on September 05,2022 at 01:24:51 UTC from IEEE Xplore. Restrictions apply.

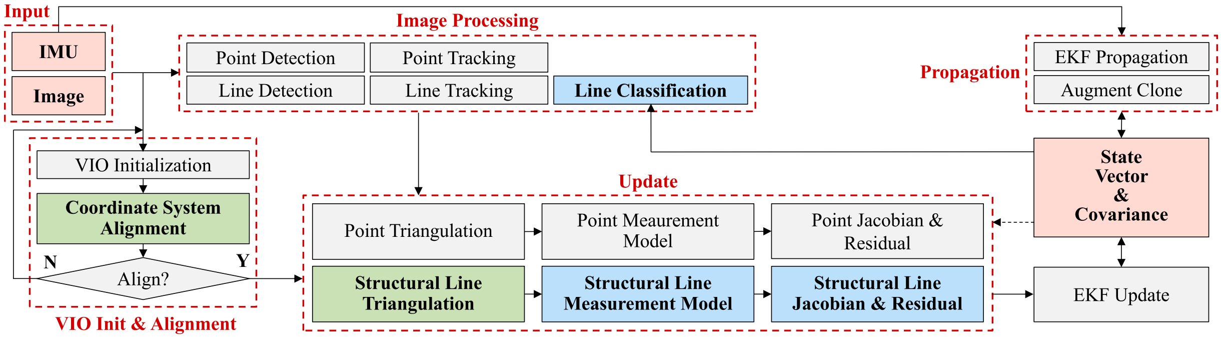


Fig. 2. Overview of the proposed VIO system. Our system is built upon OpenVINS [1] and partly based on our previous work [2]. The blue-filled boxes represent the processing steps of structural line features, and the green-filled boxes illustrate the proposed coordinate alignment and structural line triangulation algorithms.

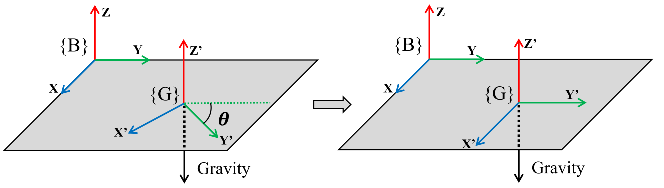


Fig. 3. Illustration of the proposed coordinate alignment method, where X Y Z are three axes of the building frame {B}, X' Y' Z' are three axes of the global frame {G}. The picture on the left shows the global coordinate {G} after traditional VIO initialization, where there is an arbitrary angle θ between X' and X, Y' and Y, only Z' axis is aligned with Z. The picture on the right shows the coordinate system after the proposed coordinate system alignment step, where all three axes of {G} are aligned with the building frame {B}.

need to ensure $\varphi_x \approx \varphi_y$. Last, the initial value of θ is obtained as:

$$\theta_{init} = \frac{1}{2}\left(\varphi_x + \varphi_y\right). \tag{3}$$

Further, we refine θ_{init} and reduce random errors by structural regularities of multiple frames. In practice, we use 11 frames for refinement and the refined rotation θ_{avg} is taking the average value of all the obtained θ_{init}. Then, we can get the rotation matrix between {B} and {G} as follows:

$${}^{G}_{B}\mathbf{R}^{T} = \begin{bmatrix} \cos\theta_{avg} & \sin\theta_{avg} & 0 \\ -\sin\theta_{avg} & \cos\theta_{avg} & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{4}$$

Lastly, we transform the initial global frame of VIO to the building frame by ${}^{G}_{B}\mathbf{R}$, which means all the state vectors and covariance of the system are changed from {G} to {B}.

The proposed coordinate alignment method requires at least two dominant directions of Manhattan world to be included in the input images, and each dominant direction must has at least two line segments. If the conditions above are not fulfilled, the proposed coordinate alignment approach may fail. In addition, the bad results of the VPs detection algorithm in [27] will cause the failure of the coordinate alignment. In that case, structural lines in Z direction still can be used in state estimation if they exist, otherwise, the system will degenerate into a point-based system. Besides, it takes approximately 2 to 3 seconds from the start to complete coordinate alignment.As shown in Fig. 4, we project three axes of the VIO's new global frame to images, which demonstrates the consistency with the true directions.

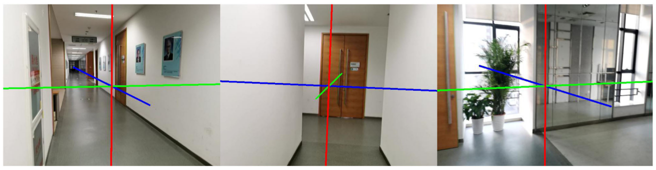


Fig. 4. Three axes of the new global frame (after the alignment) are projected onto the image planes, which illustrates that the global frame {G} is aligned with the building frame {B}. In the figures, blue, green, and red lines denote the projections of the X-axis, Y-axis, and Z-axis, respectively.

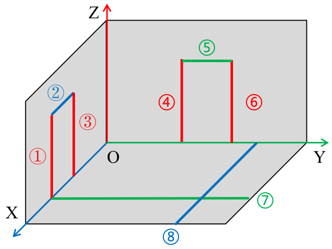


Fig. 5. Illustration of a Manhattan world scene. Blue, green, and red lines are structural lines in X, Y and Z directions, respectively.

V. Structural Line Processing

A. Structural Line Parametrization

A structural line with X-direction can be intersected with YZ-plane at a point. The point has 2D coordinates in YZ-plane. The 2D coordinates are used as parameter representations of this structural line, which is termed as projection coordinate of this structural line. YZ-plane is also called the projection plane of this structure line. These are similar to structural lines with Y and Z directions. Examples are shown in Fig. 5. Further, we use the First-Estimates Jacobian [29] technology to avoid inconsistencies in the proposed system since the structural line features are parameterized in the global frame.

B. Line Classification

The purpose of this part is to judge whether an extracted line is a structural line and classify the structural line further. The Manhattan world's three VPs can be calculated as:

$$\mathbf{v}_i = \mathbf{K}^{C}_{I}\mathbf{R}^{I}_{G}\mathbf{R}\mathbf{e}_i, \quad i = x, y, z, \tag{5}$$

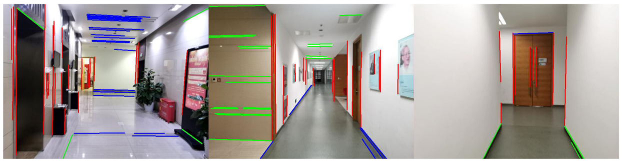


Fig. 6. Structural line classification results on Weak-texture Easy sequence of the self-collected dataset, where blue, green, and red lines are the classified structural lines in X, Y and Z directions.

where ${}^C_I\mathbf{R}$ and ${}^I_G\mathbf{R}$ denote the rotation from the IMU frame $\{\text{I}\}$ to the camera frame $\{\text{C}\}$, and the one from $\{\text{G}\}$ to $\{\text{I}\}$, respectively.

To determine a detected line's direction, we draw a ray from each $\mathbf{v}_i$ to the middle point of the detected line and check angles between the rays and the detected line. If all three angles with X, Y, Z directions are beyond the angle threshold (set as 3 degrees in practice), the detected line is not a structural line. If one of the three angles is less than an angle threshold, we determine the detected line as a structural line in the corresponding VP's direction. For example, if a detected line $\mathbf{l}_j$ has the middle point $\mathbf{m}_j$, the angle between $\mathbf{l}_j$ and the line segment connecting $\mathbf{v}_x$ and $\mathbf{m}_j$ is less than the angle threshold, we recognize $\mathbf{l}_j$ as a structural line in X direction. In particular, if more than one angle is less than the angle threshold, we choose the direction with the smallest angle.

Further, we utilize direction consistency checks in multi-frames to improve the recognition accuracy of structural lines. A structural line $\mathbf{L}$ is observed in n frames from firstly being extracted to tracking failure. Due to errors in the detection and classification procedures, $\mathbf{L}$ may be classified into different directions in different frames. Assuming most of the frames (k frames) are classified $\mathbf{L}$ into X direction, only when k/n is greater than 0.75, we will recognize $\mathbf{L}$ as the structural line in X direction. Otherwise, $\mathbf{L}$ is determined as a non-structural line. Fig. 6 shows the classification results of the structural lines.

C. Measurement Model and Jacobians

Before computing the reprojection errors, we should construct Plücker coordinates of the structural lines from their projection coordinates first, and then calculate their projections on image planes. This paper adopts a projective line measurement model based on our previous work [2]. The difference is, we analytically derive Jacobians and reprojection errors with respect to the structural lines under a new global frame.

For a structural line $\mathbf{L}$, let $\mathbf{q}$ be its 2D parameterized projection coordinate in space, and $\mathbf{Q}$ be the intersection of $\mathbf{L}$ and its projection plane in space $\{\text{G}\}$. We can get $\mathbf{Q}$ as follows:

$$ {}^G\mathbf{Q} = \mathbf{P}^T\mathbf{q}, \tag{6}$$

where $\mathbf{P}$ is the transformation matrix from $\mathbf{q}$ to $\mathbf{Q}$:

$$\mathbf{P} = \begin{cases} \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} & \text{if } \mathbf{L} \text{ in X direction} \\ \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} & \text{if } \mathbf{L} \text{ in Y direction} \\ \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} & \text{if } \mathbf{L} \text{ in Z direction} \end{cases}. \tag{7}$$

We can construct the Plücker coordinate of $\mathbf{L}$ as follows:

$$ {}^G\mathbf{L} = \begin{bmatrix} {}^G\mathbf{n} \\ {}^G\mathbf{d} \end{bmatrix} = \begin{bmatrix} \lfloor {}^G\mathbf{Q} \rfloor_\times {}^G\mathbf{d} \\ {}^G\mathbf{d} \end{bmatrix} = \begin{bmatrix} -\lfloor {}^G\mathbf{d} \rfloor_\times {}^G\mathbf{Q} \\ {}^G\mathbf{d} \end{bmatrix}, \tag{8}$$

where $\mathbf{n}$ is the normal vector of the plane containing $\mathbf{L}$ and the origin O, and $\mathbf{d} = \mathbf{e}_i,\ i = x, y, z$ represents the direction of $\mathbf{L}$, which can be obtained after the line classification step.

Then we project $\mathbf{L}$ from $\{\text{G}\}$ to camera frame $\{\text{C}\}$:

$$ {}^C\mathbf{L} = \begin{bmatrix} {}^C\mathbf{n} \\ {}^C\mathbf{v} \end{bmatrix} = {}^C_G\mathbf{T}\,{}^G\mathbf{L} = \begin{bmatrix} {}^C_G\mathbf{R} & \lfloor {}^G\mathbf{t}_C \rfloor_{\times G}^{\ C}\mathbf{R} \\ \mathbf{0} & {}^C_G\mathbf{R} \end{bmatrix} {}^G\mathbf{L}, \tag{9}$$

where ${}^C_G\mathbf{R}$ and ${}^G\mathbf{t}_C$ are the rotation matrix and the translation vector from $\{G\}$ to $\{C\}$. Last, we project ${}^C\mathbf{L}$ to image plane as:

$$ \mathbf{l} = \begin{bmatrix} l_1 & l_2 & l_3 \end{bmatrix}^T = \begin{bmatrix} f_y & 0 & 0 \\ 0 & f_x & 0 \\ -f_y c_x & -f_x c_y & f_x f_y \end{bmatrix} \mathbf{n}_c, \tag{10}$$

where f_x, f_y, c_x, c_y are the intrinsic parameters of camera. We calculate the reprojection error $\mathbf{e}_l$ of $\mathbf{L}$ by the point-to-line distance between the projected line $\mathbf{l}$ and the tracked lines' endpoints $\mathbf{x}_s$ and $\mathbf{x}_e$:

$$ \mathbf{e}_l = \begin{bmatrix} d(\mathbf{x}_s, \mathbf{l}) \\ d(\mathbf{x}_e, \mathbf{l}) \end{bmatrix} = \begin{bmatrix} \frac{\mathbf{x}_s^T \mathbf{l}}{\sqrt{l_1^2 + l_2^2}} \\ \frac{\mathbf{x}_e^T \mathbf{l}}{\sqrt{l_1^2 + l_2^2}} \end{bmatrix}, \tag{11}$$

where $\mathbf{x}_s = [u_s, v_s, 1]^T$ and $\mathbf{x}_e = [u_e, v_e, 1]^T$ are pixel coordinates of the starting point and the ending point respectively.

We can calculate Jacobians by linearizing the above measurement model for the system state and structural line positions:

$$ \mathbf{e}_{ij} \simeq \mathbf{H}_{\mathbf{x}_{ij}} \tilde{\mathbf{x}_j} + \mathbf{H}^G_{l_{ij}} \tilde{\mathbf{l}_i} + \mathbf{n}_{ij}, \tag{12}$$

where $\mathbf{x}_j, j = 1, \ldots, m$ are camera poses, $\mathbf{l}_i, i = 1, \ldots, n$ are structural lines, $\mathbf{n}_{ij}$ denotes the noise of the line reprojection error, which is modelled as zero-mean Gaussian. $\mathbf{H}_{\mathbf{x}_{ij}}$ and $\mathbf{H}_{\mathbf{l}_{ij}}$ are Jacobians of the measurement to the camera pose and the line position, respectively.

$$ \mathbf{H}_{\mathbf{x}_{ij}} = \begin{bmatrix} \frac{\partial \mathbf{e}_l}{\partial {}^C_G\mathbf{R}} \\ \frac{\partial \mathbf{e}_l}{\partial {}^G\mathbf{t}_C} \end{bmatrix}, \mathbf{H}_{\mathbf{l}_{ij}} = \frac{\partial \mathbf{e}_l}{\partial \mathbf{q}}. \tag{13}$$

$\frac{\partial \mathbf{e}_l}{\partial \mathbf{q}}$, $\frac{\partial \mathbf{e}_l}{\partial {}^C_G\mathbf{R}}$ and $\frac{\partial \mathbf{e}_l}{\partial {}^G\mathbf{t}_C}$ can be calculated via the chain rules:

$$ \frac{\partial \mathbf{e}_l}{\partial \mathbf{q}} = \frac{\partial \mathbf{e}_l}{\partial \mathbf{l}} \frac{\partial \mathbf{l}}{\partial {}^C\mathbf{L}} \frac{\partial {}^C\mathbf{L}}{\partial {}^G\mathbf{L}} \frac{\partial {}^G\mathbf{L}}{\partial {}^G\mathbf{Q}} \frac{\partial {}^G\mathbf{Q}}{\partial \mathbf{q}} \tag{14}$$

$$ \frac{\partial \mathbf{e}_l}{\partial {}^C_G\mathbf{R}} = \frac{\partial \mathbf{e}_l}{\partial \mathbf{l}} \frac{\partial \mathbf{l}}{\partial {}^C\mathbf{L}} \frac{\partial {}^C\mathbf{L}}{\partial {}^C_G\mathbf{R}} \tag{15}$$

$$ \frac{\partial \mathbf{e}_l}{\partial {}^G\mathbf{t}_C} = \frac{\partial \mathbf{e}_l}{\partial \mathbf{l}} \frac{\partial \mathbf{l}}{\partial {}^C\mathbf{L}} \frac{\partial {}^C\mathbf{L}}{\partial {}^G\mathbf{t}_C} \tag{16}$$

Then the measurement of the structural line features can be used to update the VIO system as [2].

VI. Structural Line Triangulation

To utilize the structural line features in the proposed filtering-based VIO system, an estimation of its 3D line parameters is needed to linearize the measurement model [10]. For the structural line features, the existing triangulation algorithms

Authorized licensed use limited to: INSTITUTE OF AUTOMATION CAS. Downloaded on September 05,2022 at 01:24:51 UTC from IEEE Xplore. Restrictions apply.

cannot make full use of the prior information, which may produce inaccurate results. Compared with non-structural lines, the global direction of structural lines is known, which can be used to improve the triangulation accuracy. We propose a novel triangulation method for structural lines, which divides the triangulation process into two steps. First, we treat lines as non-structural lines to calculate their Plücker matrices as in [2] and then the 2D parameterized projection coordinates of structural lines are calculated based on their global directions. Second, the calculated projection coordinates are further optimized based on the Leverberg Marquardt (LM) algorithm.

A. Initial Projection Coordinates

This part aims to calculate the initial projection coordinates of a structural line. In Section V-A, the projection planes pass through the coordinate origin O. However, if a structural line is far from O, a small-angle deviation of its Plücker matrix will lead to a large error. Therefore, we use planes parallel to the projection planes and passing through the latest camera's position as a replacement. For instance, a structural line $\mathbf{L}$ can be observed from image C_n, where the optical center position of C_n is denoted as ${}^G\mathbf{t}_{C_n} = \begin{bmatrix} t_x & t_y & t_z \end{bmatrix}^T$. The projection plane π_P of $\mathbf{L}$ can be represented as:

$$\pi_P = \begin{cases} \begin{bmatrix} 1 & 0 & 0 & -t_x \end{bmatrix}^T & \text{if } \mathbf{L} \text{ in X direction} \\ \begin{bmatrix} 0 & 1 & 0 & -t_y \end{bmatrix}^T & \text{if } \mathbf{L} \text{ in Y direction} \\ \begin{bmatrix} 0 & 0 & 1 & -t_z \end{bmatrix}^T & \text{if } \mathbf{L} \text{ in Z direction} \end{cases}. \tag{17}$$

The 3D intersection (homogeneous coordinates) of $\mathbf{L}$ and the projection plane is calculated as:

$$\mathbf{Q} = (\mathbf{L}^*)\,\pi_P = \begin{bmatrix} X_P & Y_P & Z_P & W_P \end{bmatrix}^T, \tag{18}$$

where $\mathbf{L}^*$ is the Plücker matrix of $\mathbf{L}$. Then, we obtain the initial projection coordinates of $\mathbf{L}$ as:

$$\mathbf{q}_{init} = \frac{1}{W_P} \begin{cases} \begin{bmatrix} Y_P & Z_P \end{bmatrix}^T & \text{if } \mathbf{L} \text{ in X direction} \\ \begin{bmatrix} X_P & Z_P \end{bmatrix}^T & \text{if } \mathbf{L} \text{ in Y direction} \\ \begin{bmatrix} X_P & Y_P \end{bmatrix}^T & \text{if } \mathbf{L} \text{ in Z direction} \end{cases} \tag{19}$$

B. Projection Coordinate Refinement

The above obtained initial projection coordinates can be further refined to a more precise position by a nonlinear optimization method. In the optimization process, the global orientation of a structural line is used to refine its projection coordinates. We define the energy function as follows:

$$\mathbf{q}_{refined} = \underset{\mathbf{q}}{argmin} \frac{1}{2} \left\| \mathbf{e}_l \right\|^2, \tag{20}$$

where $\mathbf{e}_l$ is the reprojection error of $\mathbf{L}$, and the calculation of Jacobians has been presented in Section V-C. In practice, we use LM algorithm to solve (20).

VII. Experiments

In this section, we evaluate the proposed system on EuRoC MAV dataset [30] and a self-collected dataset. Three SOTA point-based VIO/SLAM systems (VINS-Mono [3], ORB-SLAM3 [27], OpenVINS [1]), two point-line-based systems (PL-VINS [4], Wei et al. [31]) and a structural line-based system (StructVIO [5]) are selected for comparisons. Specially, for ORB-SLAM3 [27] and OpenVINS [1], we use their monocular-inertial configurations. Besides, StructVIO is evaluated on both EuRoC and the self-collected dataset by using its binary executable [5]. For all the systems, we disable the loop closing module to only compare their odometry performances. The root mean squared error (RMSE) of the absolute trajectory error (ATE) is chosen as the evaluation metric. To reduce random errors, we run each system five times and report the median of results for each sequence. All experiments have been run on a desktop with Intel Core i7-CPU, at 3.2 GHz, with 32 GB RAM.

A. EuRoC MAV Dataset

EuRoC MAV dataset [30] is recorded in a large industrial machine hall and two different Vicon room environments, where there exist structural regularities. The regularities can be used to improve the accuracy of VIO system in the Manhattan world assumption. For each sequence of the datasets, it contains accurate ground truth, synchronized stereo images, and high-quality IMU measurements. We utilize IMU measurements and images from the left camera to test the proposed system.

The performance comparison results on EuRoC datasets are shown in Table I. First, ORB-SLAM3 [27] is more accurate than other methods, which is benefited from the local map refinement and the global bundle adjustment steps. Another reason is that the trajectories of ORB-SLAM are not saved in real-time, which means the historical trajectories are optimized constantly. Instead, other algorithms are saving trajectories in real-time. Second, OpenVINS is more accurate than VINS-Mono and PL-VINS. Our method is built upon OpenVINS, and the proposed novelties make our method more accurate than OpenVINS. Therefore, our method obtains higher precision than VINS-Mono and PL-VINS. Third, both StructVIO and OpenVINS are MSCKF-based methods, our method gets higher accuracy than StructVIO, which is benefited from the improvement of OpenVINS and the proposed novelties. Besides, we can conclude that point-line-based systems are more accurate than point-based systems (PL-VINS [4] versus VINS-Mono [3], Wei et al. [2] versus OpenVINS [1]). More importantly, the proposed point and structural line based VIO obtains more accurate trajectories than other systems except for ORB-SLAM3, which demonstrates the effectiveness of structural regularities.

Compared with OpenVINS, we find that the proposed system in the Machine Hall scene achieves more significant accuracy improvement than Vicon Room scene. The principal reason is that the Machine Hall scene is more consistent with the Manhattan world assumption, while Vicon Room scene has lines in various directions, which may lead to the wrong classification results of structural lines.

In addition, we conduct ablation experiments to evaluate the proposed coordinate alignment and structural line triangulation algorithms. Since both the coordinate alignment and the structural line triangulation algorithms are indispensable for the proposed system, we only disable a certain step to illustrate their importance. In Table I, **Ours w/o Align.** represents the proposed system in which only one image frame is used to calculate θ and **Ours w/o Opti.** means the system uses the structural line triangulation algorithm without LM optimization. We can observe that no matter which step is disabled, the accuracy will decline, which proves the effectiveness of the proposed algorithms further. When only one image frame is used

Authorized licensed use limited to: INSTITUTE OF AUTOMATION CAS. Downloaded on September 05,2022 at 01:24:51 UTC from IEEE Xplore. Restrictions apply.

TABLE I
PERFORMANCE COMPARISON ON EUROC DATASET (RMSE ATE IN METER)

Dataset	MH_01	MH_02	MH_03	MH_04	MH_05	V1_01	V1_02	V1_03	V2_01	V2_02	V2_03
ORB-SLAM3 [27]	**0.019**^{1}	**0.040**^{1}	**0.038**^{1}	**0.145**^{1}	**0.076**^{1}	**0.044**^{1}	**0.015**^{1}	**0.047**^{1}	**0.045**^{1}	**0.024**^{1}	**0.078**^{1}
VINS-Mono [3]	0.155	0.178	0.195	0.348	0.302	0.089	0.111	0.188	0.086	0.158	0.278
PL-VINS [4]	0.157	0.170	0.227	0.303	0.282	0.070	0.123	0.180	0.081	0.116	0.277
OpenVINS [1]	0.085	0.144	0.104	0.273	0.275	0.063	0.087	**0.072**^{2}	0.111	0.073	0.222
StructVIO [5]	0.103	0.080	0.141	0.144	0.261	0.083	0.088	0.102	**0.058**	**0.061**	0.166
Wei et al. [2]	0.071	0.123	0.108	0.241	0.267	**0.057**^{2}	0.082	0.073	0.115	0.064	0.144
Ours w/o Align.	0.075	0.092	0.106	0.162	0.265	0.090	0.097	0.090	0.106	0.092	0.176
Ours w/o Opti.	0.080	0.088	0.093	0.143	0.259	0.087	0.094	0.086	0.094	0.090	0.158
Ours	**0.069**^{2}	**0.074**^{2}	**0.091**^{2}	**0.136**^{2}	**0.251**^{2}	0.084	**0.079**^{2}	0.085	0.095	0.087	**0.143**^{2}

^{1} and ^{2} mean the highest and second highest accuracy among all algorithms on the same sequence.
All Results are Obtained by Our Device While Keeping Their Default Parameter Configuration.

TABLE II
PERFORMANCE COMPARISON ON THE SELF-COLLECTED DATASET (RMSE ATE IN METER)

Dataset	ORB-SLAM3	VINS-Mono	PL-VINS	OpenVINS	Wei et al.	StructVIO	w/o Align.	w/o Opti.	Ours
Weak-texture Easy (W_E)	—	2.976	5.079	1.198	0.979	1.091	1.711	0.542	**0.353**
Weak-texture Hard (W_H)	—	1.690	1.784	1.037	0.755	—	1.573	0.681	**0.609**
Stairs Easy (S_E)	0.563	1.077	0.474	0.337	0.108	1.138	0.180	0.187	**0.097**
Stairs Hard (S_H)	0.862	0.803	1.332	0.501	0.229	0.807	0.435	0.207	**0.168**

to calculate θ (w/o Align.), the error increases by 0.014 meters on average. The reason is that the coordinate alignment based on a single image is too uncertain, which may lead to failure of the proposed coordinate alignment processing. In that case, structural lines in Z direction can still be used in state estimation.

B. *Self-Collected Dataset*

Furthermore, we test the proposed system on a self-collected dataset. The self-collected dataset is recorded in two different indoor scenes, named Weak-texture and Stairs. Notably, both scenes strictly satisfy the Manhattan world assumption. In each scene, we hold a mobile phone (HUAWEI P30 Pro) moving along a planned route, meanwhile, the IMU measurements and monocular images are recorded. The frequencies of IMU and camera are 200 Hz and 20 Hz, respectively. The resolution of the recorded image is 640×480, and the planned route is about 138 meters.

In each scene, we collect two sequences, named Easy and Hard, based on different movement patterns. Both sequences have challenging scenes such as weak texture scenes and motion blur situations. Similar to [5], we use a printed ArUco marker [32] to get ground truth at the beginning and the end for all sequences. As a result, the accumulated drift of the whole trajectory can be measured, which is considered as our quantitative evaluation metric.

We compare the proposed VIO system with the three point-based and two point-line-based algorithms, the results are shown in Table II. First, because the Weak-texture scene exists corridors with almost pure white walls, ORB-SLAM3 fails in the scene. ORB-SLAM3 [27] runs successfully in the Stairs scene, but achieves worse accuracy. Compared with ORB-SLAM3, Wei et al. [2] and our methods obtain more accurate trajectories benefiting from the combination of point and line features. Second, our methods integrate structural line features, and Wei et al. divide line features as "MSCKF" lines and "SLAM" lines based on OpenVINS. Compared with OpenVINS, the error of our method decreases by about 0.461 meters, and Wei et al. decreases by about 0.250 meters. The reason is that the usage method of line features in Wei et al. may lead to a large number of unstable line segments added to the system, which will corrupt the system in some cases. In another word, structural line features provide more strict geometry constraints for VIO system compared to nonstructured lines, and their physical properties can ensure a high inliers rate, which is helpful for the VIO system. Compared with StructVIO, our method achieves higher accuracy, which is benefited from the proposed novelties.

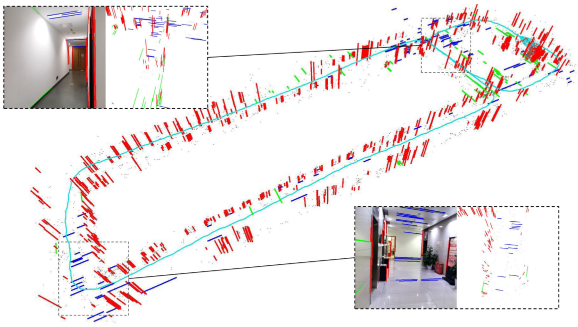
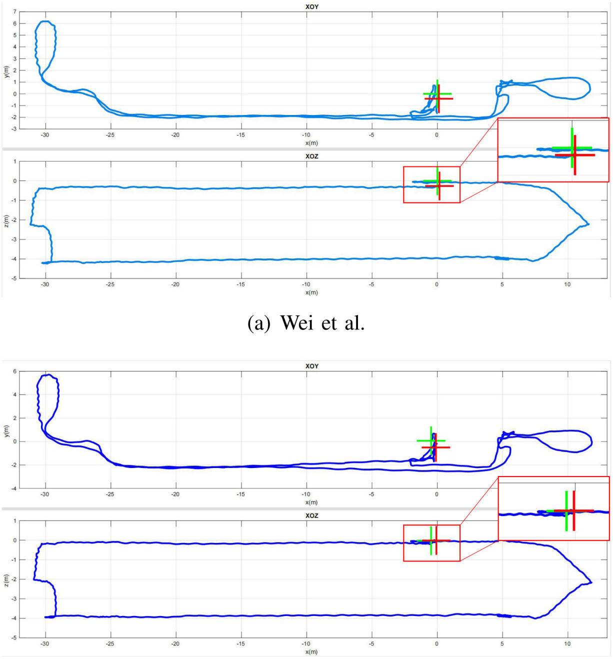


Fig. 7. Results of the proposed VIO system on Weak-texture Easy sequence of the self-collected datasets.

Specially, we also conduct ablation experiments on the self-collected datasets. As shown in Table II, no matter which step is disabled, the accuracy of the system will decline sharply. In particular, if we disable the LM optimization of the structural line triangulation algorithm (w/o Opti.), the error will increase by about 0.097 meters, which proves that the global orientation of structural lines can be used to refine their 3D position. Besides, when only one image frame was used to calculate θ (w/o Align.), the error of the system increases by about 0.669 meters on average due to the failure of the proposed coordinate alignment method. The reason is that the inaccurate alignment result leads to a worse system initialization state.

Fig. 7 shows the structural line map obtained by the proposed point and structural line based VIO system on Weak-texture

Authorized licensed use limited to: INSTITUTE OF AUTOMATION CAS. Downloaded on September 05,2022 at 01:24:51 UTC from IEEE Xplore. Restrictions apply.



(a) Wei et al.

(b) Ours

Fig. 8. Trajectories of Wei et al. [2] and the proposed system from Stairs Hard sequence of the self-collected datasets. When capturing the videos, a man holding a phone traveled from the second floor to the first floor and returned to the start position finally. The projections of trajectories on XOZ and XOY are shown. The green '+' is the starting point and the red '+' is the ending point. The red boxes show the proposed method has less drift in Z direction, which is benefited from the structural regularities.

TABLE III
EXECUTION TIME COMPARISON (IN MILLISECONDS)

Operation	PL-VINS [4]	Wei et al. [2]	Ours
Point Detection & Tracking	8.90	3.30	3.34
Line Detection & Tracking	30.20	17.59	17.60
Structural Line Classification	—	—	0.01
Line Triangulation	0.05	0.04	0.16
Optimization / EKF	22.05	8.15	3.83
Total	61.20	29.08	**24.94**

Easy sequence. The line map is sparse since the scene has fewer textures. We can also observe that the structural lines in X and Y directions (marked in blue and green in the figure) are less than the structural lines in Z direction (marked in red). There are two reasons for the phenomenon. First, there are fewer lines in the X and Y directions in the scene. Second, the lines in X and Y directions are hard to be triangulated due to degradation problems [10]. Fig. 8 shows the trajectories of Wei et al. [2] and ours in Stairs Hard sequence. Interestingly, our method has less drift in Z direction benefited from the structural regularities.

C. Runtime Evaluation

Lastly, we compare the computation time by the proposed system, PL-VINS [4] and our previous work [2] (Wei et al.). We run them on Weak-texture Hard sequence and record the average time waste of each step. The results are shown in Table III. PL-VINS has the lowest execution efficiency because of the time-consuming feature detection and tracking algorithms and the optimization-based framework. Wei et al. [2] and the proposed system are both filtering-based methods, which are more efficient than the optimization-based system. Besides, both Wei et al. and our methods utilize an efficient line tracking algorithm [31]. In particular, this paper takes structural lines as MSCKF features, which saves time for the MSCKF and SLAM features classification and the state update step. Above all, the proposed system is highly efficient.

For the coordinate alignment step, we use the method in [28] to detect vanishing points first. The VP detection algorithm needs 82.8 ms to process an image averagely, which is very time-consuming. However, the alignment step is only executed once after system initialization, which has little influence on the system. In particular, the other steps of coordinate alignment only need 0.658 ms, which proves the efficiency of the proposed coordinate alignment algorithm.

VIII. CONCLUSION

This paper presents a new point and structural line based VIO system to utilize the structural regularities in a Manhattan world environment. For using structural information easily, we propose a novel coordinate alignment algorithm. Then, the forms of 2D parameterization, Jacobians, and reprojection errors become much more concise by our derivations. Besides, we also propose a new structural line triangulation method, in which the global orientations of structural lines are used to refine their 3D positions. All the novelties make an efficient and accurate VIO system, which can also output a geometrical line map. The proposed VIO system is evaluated on EuRoC MAV datasets and a self-collected dataset. Experimental results prove that the proposed method achieves better accuracy compared with the state-of-the-art point-based systems (VINS-Mono [3] and OpenVINS [1]), point-line-based systems (PL-VINS [4] and Wei et al. [2]), and structural line based system (StructVIO [5]), which is benefited from the proposed novelties. Specifically, the self-collected dataset is recorded on strict Manhattan world scenes. In the dataset, our method improves about 40.7% compared with our previous work [2] in terms of accuracy. In particular, our method is the most efficient among the point-line-based systems.

However, the proposed system also has two main limitations. First, Manhattan world assumption makes our system can only be used in indoor or urban scenes. Second, the proposed coordinate alignment method requires at least two dominant directions of Manhattan world included in the images, and each dominant direction must have at least two line segments. If the conditions above are not fulfilled, this will fail. In that case, structural lines in Z direction still can be used if they exist. Otherwise, the system will degenerate into a point-based system. In the future, we plan to look at how to minimize the problem with Manhattan world assumption. We are also interested in using structural line features to build structural planes and create geometrical maps.

REFERENCES

[1] 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.

[2] H. Wei, F. Tang, Z. Xu, C. Zhang, and Y. Wu, "A point-line vio system with novel feature hybrids and with novel line predicting-matching," *IEEE Robot. Automat. Lett.*, vol. 6, no. 4, pp. 8681–8688, Oct. 2021.

[3] 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.

Authorized licensed use limited to: INSTITUTE OF AUTOMATION CAS. Downloaded on September 05,2022 at 01:24:51 UTC from IEEE Xplore. Restrictions apply.

[4] 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*.

[5] 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.

[6] Y. Wu, F. Tang, and H. Li, "Image-based camera localization: An overview," *Vis. Comput. Ind., Biomed., Art*, vol. 1, no. 1, pp. 1–13, 2018.

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

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

[9] 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, 2018, Art. no. 1159.

[10] Y. Yang, P. Geneva, K. Eckenhoff, and G. Huang, "Visual-inertial odometry with point and line features," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2019, pp. 2447–2454.

[11] Y. Li, N. Brasch, Y. Wang, N. Navab, and F. Tombari, "Structure-SLAM: Low-drift monocular SLAM in indoor environments," *IEEE Robot. Automat. Lett.*, vol. 5, no. 4, pp. 6583–6590, Oct. 2020.

[12] H. Li, J. Yao, J.-C. Bazin, X. Lu, Y. Xing, and K. Liu, "A monocular SLAM system leveraging structural regularity in manhattan world," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2018, pp. 2518–2525.

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

[14] A. I. Mourikis and S. I. Roumeliotis, "A multi-state constraint Kalman filter for vision-aided inertial navigation," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2007, pp. 3565–3572.

[15] G. Zhang and I. H. Suh, "Building a partial 3D line-based map using a monocular SLAM," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2011, pp. 1497–1502.

[16] G. Zhang and I. H. Suh, "A vertical and floor line-based monocular SLAM system for corridor environments," *Int. J. Control, Automat. Syst.*, vol. 10, no. 3, pp. 547–557, 2012.

[17] J. Lee and S.-Y. Park, "PLF-VINS: Real-time monocular visual-inertial SLAM with point-line fusion and parallel-line fusion," *IEEE Robot. Automat. Lett.*, vol. 6, no. 4, pp. 7033–7040, Oct. 2021.

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

[19] G. Schindler and F. Dellaert, "Atlanta world: An expectation maximization framework for simultaneous low-level edge grouping and camera calibration in complex man-made environments," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, 2004, vol. 1, pp. I–I.

[20] H. Li, J. Zhao, J.-C. Bazin, and Y.-H. Liu, "Quasi-globally optimal and near/true real-time vanishing point estimation in manhattan world," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 3, pp. 1503–1518, Mar. 2022.

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

[22] K. Joo, P. Kim, M. Hebert, I. S. Kweon, and H. J. Kim, "Linear RGB-D SLAM for structured environments," *IEEE Trans. Pattern Anal. Mach. Intell.*, early access, 2021, doi: 10.1109/TPAMI.2021.3106820.

[23] J. Liu and Z. Meng, "Visual SLAM with drift-free rotation estimation in manhattan world," *IEEE Robot. Automat. Lett.*, vol. 5, no. 4, pp. 6512–6519, Oct. 2020.

[24] H. Li, J. Zhao, J.-C. Bazin, W. Chen, K. Chen, and Y.-H. Liu, "Line-based absolute and relative camera pose estimation in structured environments," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2019, pp. 6914–6920.

[25] D. G. Kottas and S. I. Roumeliotis, "Exploiting urban scenes for vision-aided inertial navigation," in *Proc. Robot.: Sci. Syst.*, 2013, pp. 24–28.

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

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

[28] 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 *Proc. IEEE Winter Conf. Appl. Comput. Vis.*, 2017, pp. 345–353.

[29] G. P. Huang, A. I. Mourikis, and S. I. Roumeliotis, "A first-estimates Jacobian EKF for improving SLAM consistency," in *Experimental Robotics*. Berlin, Heidelberg: Springer, 2009, pp. 373–382.

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

[31] H. Wei, F. Tang, C. Zhang, and Y. Wu, "Highly efficient line segment tracking with an IMU-KLT prediction and a convex geometric distance minimization," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2021, pp. 3999–4005.

[32] F. J. Romero-Ramirez, R. Muñoz-Salinas, and R. Medina-Carnicer, "Speeded up detection of squared fiducial markers," *Image Vis. Comput.*, vol. 76, pp. 38–47, 2018.

Authorized licensed use limited to: INSTITUTE OF AUTOMATION CAS. Downloaded on September 05,2022 at 01:24:51 UTC from IEEE Xplore. Restrictions apply.