S-VIO: Exploiting Structural Constraints for RGB-D Visual Inertial Odometry

Pengfei Gu and Ziyang Meng, *Senior Member, IEEE*

***Abstract*—Vision-based localization is an essential problem for autonomous systems while the performance of vision-based odometry degrades in the challenging scenario. This letter presents S-VIO, an RGB-D visual inertial odometry (VIO) which fully uses multi-sensor measurements (i.e. depth, RGB and IMU), heterogeneous landmarks (i.e. points, lines and planes) and structural regularity of the environment to obtain a robust and accurate localization result. In order to detect the underlying structural regularity of the environment, a two-step Atlanta world inference method is proposed. Leveraging the gravity direction estimated by the VIO system, the proposed algorithm first generates horizontal Atlanta axis hypotheses from a set of recently optimized plane landmarks. Then the following plane landmarks and line clusters are used to filter out the occasionally observed axes based on the persistence of the observation. The remaining axes will survive and be saved in the Atlanta map for future re-observations. Particularly, an efficient mined-and-stabbed (MnS) method is applied to classify the structural line and extract the vanishing point from each line cluster. In addition, a closed-form initialization method for the structural line feature is proposed, which leverages the known direction to obtain a better initial estimation. Integrated with the above novelties, S-VIO is tested on two public real-world RGB-D inertial datasets. Experiments demonstrate that S-VIO has better accuracy and robustness compared to state-of-the-art VIO and RGB-D VIO algorithms.**

***Index Terms*—SLAM, Localization.**

I. Introduction

LOCALIZATION in unknown environment is a fundamental technology for many robotics applications, such as autonomous navigation and AR/VR. Due to its importance, lots of works have been conducted on the visual inertial odometry (VIO), a key part of a more sophisticated system generally named as visual inertial simultaneous localization and mapping (VI-SLAM). Compared with the full VI-SLAM system, VIO alone provides a more lightweight solution without loop closure, which is critical for real-time application on embedded devices, such as micro aerial vehicles (MAVs) and AR devices.

Nowadays, many excellent VIO/SLAM algorithms have been proposed and are shown to achieve high localization accuracy [1], [2], [3], [4]. However, their accuracy decreases in challenging environments. Typical challenging factors include weak textures, fast motion, poor illumination and dynamic objects, which may lead to large odometry drift or a complete system failure. The main issue is that in these challenging scenes, the algorithm cannot find sufficient salient features for pose estimation. Few point features can be extracted and properly tracked in the low-textured/bad-illuminated environment or at the presence of fast motion, while in dynamic scenes, the static points are mixed up with the dynamic points, negatively affecting the accuracy and robustness.

Some researchers propose to integrate the depth information from the RGB-D camera to improve the performance of pure VIO/VI-SLAM systems. It is shown that the depth information benefits both the initialization and the optimization processes of the VIO. Other researchers leverage line and planar features to improve the robustness in low-textured environments. In addition, some works utilize the structural regularity of the man-made environment to correct the rotation drift. The above works have shown better robustness and accuracy than the traditional point-based VIO/VI-SLAM systems do. However, despite the widespread use of the RGB-D VI sensor suite, there are still few works that fully exploit the information provided by such a class of sensor suite.

Motivated by the above understanding, this work proposes to exploit the benefits of heterogeneous landmarks (i.e. points, lines and planes) in combination with the structural regularity (i.e. the Atlanta world assumption) in an RGB-D inertial setting. The measurements from different modalities (i.e. depth, RGB and IMU) are fully used and fused in a tightly-coupled manner to improve the performance. Also, the coincidence between the vertical dominant direction and the gravity direction is leveraged to facilitate the inference of the Atlanta world.

In summary, the main contributions in this letter are:

- A first open-sourced tightly-coupled RGB-D VIO algorithm combining point, line, planar landmarks and structural regularity of the environment to achieve high localization precision and robustness.
- A two-step Atlanta world inference method leveraging the estimated gravity direction and sequential information to detect stable dominant directions from a sequence of RGB-D images.
- A closed-form structural line initialization method leveraging the known direction of the line to obtain a better initial estimation.

Manuscript received 15 November 2022; accepted 5 April 2023. Date of publication 25 April 2023; date of current version 1 May 2023. This letter was recommended for publication by Associate Editor F. Pomerleau and Editor S. Behnke upon evaluation of the reviewers' comments. This work was supported in part by Beijing Natural Science Foundation under Grant JQ20013 and in part by the National Natural Science Foundation of China under Grants 61833009, 62273195, and U19B2029. *(Corresponding author: Ziyang Meng.)*

The authors are with the Department of Precision Instrument, Tsinghua University, Beijing 100084, China (e-mail: gpf20@mails.tsinghua.edu.cn; ziyang-meng@mail.tsinghua.edu.cn).

The code and the supplementary material will be available at https://github.com/oranfire/SVIO.

Digital Object Identifier 10.1109/LRA.2023.3270033

Additionally, as a part of the proposed Atlanta world inference method, we implement an improved mined-and-stabbed algorithm for structural line clustering. Different from the original algorithm [5], our implementation avoids the need for cumbersome coefficient calculation and high degree algebraic equation solving, and thus provides a precise yet easy-to-compute solution. Finally, the proposed odometry, namely S-VIO is evaluated on two real-world RGB-D inertial datasets and compared with state-of-the-art VIO and RGB-D VIO algorithms to demonstrate its superiority in robustness and accuracy.

II. Related Works

A. RGB-D Inertial SLAM

In the past decades, with the widespread use of low-cost commodity RGB-D cameras, real-time RGB-D SLAM and dense reconstruction algorithms have been extensively studied. Nevertheless, pure RGB-D SLAM approaches lack robustness for aggressive camera motion, low texture and geometric variation, thus motivating the study on the RGB-D inertial SLAM [6], [7], [8], [9], [10], [11], [12].

The first tightly-coupled dense RGB-D inertial SLAM is proposed by Laidlow et al. [6]. They extend a classical RGB-D SLAM system with tightly-coupled IMU integration. In [7], Shan et al. extend a VI-SLAM system, VINS-Mono [2] with the depth information. Sparse depth measurements on point features are incorporated to improve the accuracy of the VIO algorithm. Zhang et al. [8] also propose a sparse RGB-D VIO system built on VINS-Mono. Different from [7], they establish the uncertainty-aware constraint based on the Gaussian mixture model (GMM) and propose a hybrid PnP method for depth-aided VIO initialization. Some researchers also utilize the depth image to facilitate localization in dynamic scenarios. Liu et al. [9] segment dynamic objects on the depth image with the help of a lightweight object detection model. This method is integrated into a VI-SLAM system, Dynamic-VINS, supporting real-time processing on embedded platforms. In the second place, planar features are also extracted and exploited from dense depth image. For example, Hsiao et al. [10] propose DPI-SLAM, where planes are extracted from keyframes' depth images and optimized together with poses, velocities and IMU biases in a global factor graph. Yang et al. [11] exploit planar landmarks in a filter-based VIO. Additionally, planar points are distinguished from non-planar points to enforce the point-on-plane constraint. Recently, Chen et al. [12] propose a tightly-coupled RGB-D inertial SLAM system, VIP-SLAM, which leverages the point and planar information. In this letter, we further utilize the combination of point, line and planar features, as well as the structural regularity of the environment to benefit the localization in an RGB-D inertial setting.

B. Line and Plane-Aided SLAM

Most existing SLAM methods relying on points perform well in well-textured scenes, however, easily suffer from the degeneration problem in low-textured scenes.

Some works introduce the use of line and planar landmarks to address this problem [12], [13], [14], [15], [16]. Lu et al. [13] propose a heterogeneous landmark-based monocular SLAM system, where the heterogeneous visual features and their inner relationship are encoded in a multilayer feature graph (MFG) and optimized together. In [14], Gomez-Ojeda et al. present a stereo SLAM system employing the point and line features, which operates robustly in the low-textured scenario. There is also a trend combining direct methods with lines [17] and planes [18]. These methods generally minimize the photometric error term and update the points belonging to the same line/plane jointly, in which way the efficiency is improved, and the noisy estimation is avoided.

Some works also leverage the relationship between different geometrical entities to improve the accuracy and efficiency of the SLAM system. Li et al. [15] propose a novel co-planar parameterization scheme to reduce the computational cost, where co-planar points and lines are represented by their 2D image observations and the parameter of the plane. In [16], Lee et al. propose a monocular VI-SLAM system with point-line and parallel-line fusion. Similar to the above works, both the sensor measurement and the geometrical relationship of the point, line and planar features are utilized in the proposed system. Besides, we propose a closed-form initialization method for the parallel line features by incorporating the known direction information.

C. Structural SLAM

In the absence of loop closure, the majority of existing odometry methods easily suffer from large drift error after a long period running. However, the man-made environment where the robots work mostly exhibits strong structural regularity which can be leveraged to correct the rotation drift of the odometry.

Different assumptions are made to capture the structural regularity. One of the widely adopted assumptions is the Manahattan world (MW) assumption [19], which assumes that most man-made structures are parallel to three orthogonal directions. Under this assumption, several works decouple the rotational and translational motion estimation to obtain a drift-free rotation estimation [20], [21], [22], [23]. In [20], Kim et al. propose a low-drift RGB-D VO system, LPVO. It firstly estimates drift-free rotation from lines and planes by exploiting the MW assumption and then computes the translational motion by minimizing the de-rotated reprojection error. Liu et al. [21] further leverage the drift-free rotation estimation to improve the computational efficiency of the bundle adjustment. Li et al. [22] propose an RGB-D SLAM system exploiting the MW assumption. An additional fine-tuned step is applied to compensate for the general environment that does not strictly adhere to the MW assumption. Company-Corcoles et al. [23] present an RGB-D visual odometry combining point and line features. It is built on ORB-SLAM2 and estimates the Manhattan axes of the scene (if they exist) on the fly.

Another widely used assumption is the Atlanta world (AW) assumption [24]. It posits that majority of structure elements present in the environment are parallel to certain axes consisting of one vertical direction and multiple horizontal directions orthogonal to vertical direction, which is more flexible than the MW assumption. It can also be viewed as a collection of multiple local MWs with different headings. Zou et al. [25] present a

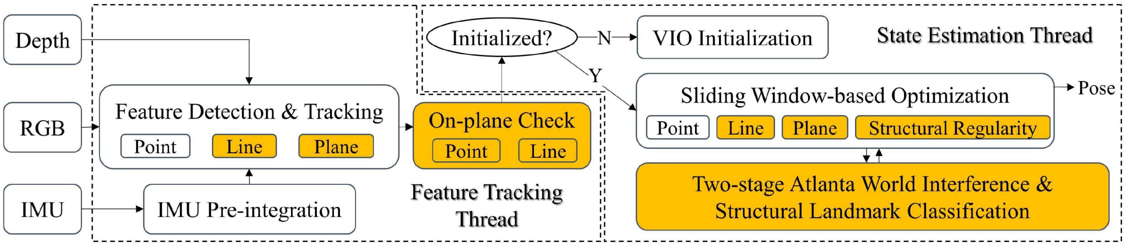


Fig. 1. The framework of S-VIO. With respect to VINS-Mono [2] and DUI-VIO [8], the newly added components are colored as orange.

filter-based VIO system leveraging the AW assumption. The local MWs are detected on the fly and the structural lines aligned with each local MW are parameterized by the 2-Dof intersection point on the projected plane. In [26], Joo et al. propose an RGB-D SLAM system utilizing the AW assumption. In this letter, we propose a two-step AW interference method from the plane and the line cluster in an RGB-D inertial setting. Specifically, an improved MnS algorithm is deployed to cluster the lines efficiently.

III. S-VIO Overview

A. System Overview

The overall framework of the proposed S-VIO system is depicted in Fig. 1. Our system takes synchronized RGB-D images and IMU measurements as input. The basic structure of S-VIO is modified on VINS-Mono [2] and DUI-VIO [8]. In particular, two main threads run parallel in S-VIO: feature tracking and state estimation. Synchronized RGB and depth images are passed to the feature tracking thread for feature detection and tracking. An on-plane check module assigns co-planar point and line features to the plane based on the plane segmentation result. Meanwhile, IMU measurements between two adjacent frames are pre-integrated. Then, the state estimation thread summarizes the feature information and IMU pre-integration results for the estimator initialization and pose estimation. For the estimator initialization part, we adopt the method proposed by DUI-VIO. For the state estimation part, a sliding window-based optimization is performed to obtain a maximum posteriori estimation (Section III-C). After the optimization, an outlier removal method identifies and discards the features with large re-projection error. Then we detect the Atlanta World model (Section IV-A) and associate features with previously detected dominant directions.

In particular, the feature tracking thread is responsible for three kinds of features: point, line and plane. For efficiency, they are detected and tracked in three parallel threads. The point features are detected by the Shi-Tomas corner detector and tracked with the Kanade-Lucas-Tomasi (KLT) optical flow method. For the line feature's detection and tracking, we implement a scheme combining the FLD [27] with the KLT-based sample point tracking. Additionally, broken line segments belonging to the same line are merged into a single long line for better estimation. Potential multiple plane regions are segmented on the depth image using the AHC algorithm [28] and then fitted into 3D plane using the least square method. For the plane matching, the mean point-to-plane distance and plane-to-plane normal angle are computed and compared with thresholds (5 cm for the distance and 5 degrees for the angle) to identify the matched plane.

B. State Variables

The state variables in our system are defined as follows:

$$\mathcal{X} = [x_0, \ldots, x_{d-1}, \lambda_0, \ldots, \lambda_{n-1}, l_0, \ldots, l_{m-1}, \pi_0, \ldots, \pi_{k-1}]$$
$$x_i = [p_i^w, v_i^w, q_i^w, b_{a_i}, b_{g_i}] \tag{1}$$

In (1), $\mathcal{X}$ is the entire set of state variables. d, n, m, k are the length of the sliding window, the number of point features, the number of structural line features and the number of plane features respectively. x_i is the IMU state at the time of capturing the i-th image in the sliding window. p_i^w, v_i^w, q_i^w are the i-th translation, velocity and orientation of IMU in the world frame. b_{a_i}, b_{g_i} are the i-th acceleration bias and gyroscope bias in the IMU body frame. λ is the inverse depth of the point feature in its first observation frame. l is the 2-Dof representation of the structural line feature in the world frame as described in Section IV-C. π is the 3-Dof closest point (CP) vector of the plane feature in the world frame [11].

C. Sliding Window-Based Optimization

A sliding window-based optimization is performed to estimate the above state variables in a tightly-coupled fusion. The overall objective function $\mathcal{L}$ is defined as follows:

$$\begin{aligned}\mathcal{L} = {} & \|r_p - H_p X\|^2 + \sum_i \left\| r_B\left(z_{b_{i+1}}^{b_i}, X\right)\right\|_{P_{b_{i+1}}^{b_i}}^2 \\ & + \sum_{\alpha,j} \rho\left(\left\| r_P\left(z_\alpha^{c_j}, X\right)\right\|_{P_p}^2\right) + \sum_{\alpha,j} \rho\left(\left\| r_{P,D}\left(z_\alpha^{d_j}, X\right)\right\|_{P_{pd}}^2\right) \\ & + \sum_{\gamma,j} \rho\left(\left\| r_L\left(z_\gamma^{c_j}, X\right)\right\|_{P_l}^2\right) + \sum_{\gamma,j} \rho\left(\left\| r_{\Pi,D}\left(z_\gamma^{d_j}, X\right)\right\|_{P_D}^2\right) \\ & + \sum_{\gamma} \rho\left(\left\| r_{\Pi,S}\left(z_\gamma, X\right)\right\|_{P_S}^2\right) + \sum_{\alpha,\gamma,j} \rho\left(\left\| r_{\Pi,P}\left(z_{\alpha,\gamma}^{c_j} X\right)\right\|_{P_p}^2\right) \\ & + \sum_{\beta,\gamma,j} \rho\left(\left\| r_{\Pi,L}\left(z_{\beta,\gamma}^{c_j}, X\right)\right\|_{P_l}^2\right)\end{aligned} \tag{2}$$

In (2), α, β, γ are the index of the point, line and plane feature respectively. $\rho(\cdot)$ is the Cauthy robust function to reduce the impact of outlier measurements. z, P are the measurement and the covariance respectively. r_p, H_p is the prior information after the marginalization [2]. r_B is the residual for IMU measurements. r_P, r_L are the re-projection errors on the image plane for point and line feature respectively. For the line feature, the distances between two end points of the observed line segment and the projected line are taken as the measurement [25]. $r_{P,D}$, $r_{\Pi,D}$ are the depth error for the point and plane feature respectively. For the point feature, we adopt the GMM-based depth error term used in [8]. For the plane feature, the point-to-plane cost proposed in [29] are weighted to incorporate the depth uncertainty. In addition, $r_{\Pi,P}$, $r_{\Pi,L}$ are the re-projection errors for the co-planar point and line feature formulated as in [15]. The accurate plane estimation is leveraged to benefit the estimation of co-planar point and line features. $r_{\Pi,S}$ is the structural error term for the plane feature. It requires that the normal of the plane is parallel/orthogonal to certain direction based on the structural type. In particular,

$$r_{\Pi,S} = \begin{cases} 1 - \frac{\pi_\gamma^T d_\gamma}{\|\pi_\gamma\|}, & \text{if plane is orthogonal to HDD} \\ 1 - \frac{\pi_\gamma^T z}{\|\pi_\gamma\|}, & \text{else if horizontal plane} \\ \frac{\pi_\gamma^T z}{\|\pi_\gamma\|}, & \text{else if vertical plane} \\ 0, & \text{else} \end{cases} \tag{3}$$

where z is a unit vector representing the gravity direction in the world frame, and d_γ is another unit vector representing the Atlanta direction orthogonal to the γ-th plane.

IV. Methodology

A. *Two-Step Atlanta World Inference*

In this section, we present a two-step method for the AW inference. Given a sequence of RGB-D images, the AW inference process aims to identify the underlying Atlanta structure of the scene, literally the dominant directions. It should be noted that the vertical dominant direction (VDD) has already been known once the VIO initialization completes. Therefore, the problem is reduced to find the horizontal dominant directions (HDDs) given the known vertical dominant direction.

Previous works detect the dominant directions from the vanishing point of line clusters, planes and the surface normal. However, they may differ in the quality of estimation. In the presence of the RGB-D camera, high-quality depth measurement in close range is available, and thus the plane estimate is accurate. On the other hand, line features appear noisy in the real-world cluttered environment and may provide ambiguous measurements at particular viewpoints. Motivated by this observation, we design a reliable approach to infer the AW in two steps. In the hypothesis step, hypothetical HDDs are firstly generated from the observed vertical planes based on the inlier maximum principle. Then in the verification step, the following planes and lines are used to verify the newly proposed hypotheses based on the persistence of the observation. Note that the occasionally observed horizontal directions are filtered out in the verification step. Therefore, the combination of two steps guarantees an accurate and efficient detection of the Atlanta dominant directions. We next present the details of the proposed two-step scheme.

1) Hypothesis Step: The process begins with the hypothesis step. A temporal window is established to store the recent observed plane features within a short period of time, among which vertical planes that are not associated to any of previous HDDs are selected for HDD detection. Specifically, we use a weighted mine-and-stab (MnS) algorithm, which is firstly proposed in [5] for structural line clustering, to detect hypothetical HDDs from a set of vertical planes' normal. The main steps of the HDD detection are presented as follows. Firstly, the angle between the normal of the plane and the x-axis is computed to generate the one-dimensional candidate interval for the possible HDD. Then, given a set of candidate intervals, the MnS algorithm is used to find the optimal sub-interval that overlaps with the most candidate intervals. We also attach each candidate interval with a weight representing its reliability, i.e. the plane's observation time. Sub-intervals with a sum of weight larger than a threshold are selected. Finally, hypothetical HDDs are generated as the center of these sub-intervals.

2) Verification Step: The verification step aims to eliminate occasionally observed hypothetical HDDs by using the plane and the line features. For each newly detected hypothetical HDD, we check its persistency based on the observation time. If a hypothetical HDD encounters a maximum length of M keyframe observation interruption before accumulating a minimum of N keyframe observations, it will be discarded ($M = 150$ and $N = 300$ in all our experiments). For each plane feature, we firstly compute the angle between the normal and the gravity direction to decide its structural type. If it is a vertical plane, we then traverse the HDDs to find the matched one with the smallest angle difference. For the line part, we use the vanishing points extracted from single-frame line observations to match with the HDD. It has three steps. First, lines parallel to the gravity vector are identified based on the angle between the normal vector of the line and the gravity direction. Then, the rest line features are classified into different clusters using the method in Section IV-B. Finally, the vanishing point of each line cluster is computed to search for the matched HDD. It is worth noting that the vanishing point of each line cluster, rather than each line's normal vector are utilized to retrieve the matched HDD, since the direction of the line may be ambiguous from a single line observation.

B. *Efficient Structural Line Clustering*

In particular, we implement an improved MnS approach for structural line clustering. The basic idea is the same as the algorithm proposed in [5] (later extended to MW in [30]). However, we show that the computational efficiency can be further improved.

The basic process of the MnS algorithm can be briefly summarized as follows (for detailed descriptions, please refer to [5]). In Fig. 2, the gravity direction is represented as a unit vector $\vec{z} = (0, 0, 1)$, and the sphere point S lying on the Gaussian sphere centered at O represents the unit normal vector $\vec{n}$ of a line observation. If it is a noise-free line parallel to some HDD, S should lie on the horizontal dominant plane π orthogonal to that HDD. However, when it comes to the noisy case in reality, S lies on a spherical cap ω. It is worthy noting that

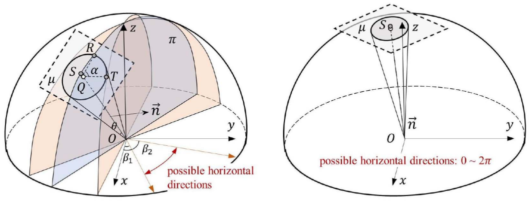
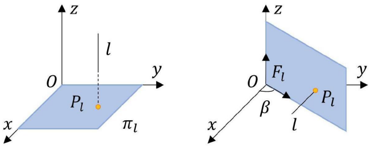

z
π
R
μ
S
α
T
Q
$\vec{n}$
θ
O
y
β_1
β_2
x
possible horizontal directions
μ
S
z
$\vec{n}$
O
y
possible horizontal directions: 0 ~ 2π
x

Fig. 2. Two situations for the candidate interval calculation. In most cases (right), the normal vectors (red arrows) of two vertical planes which are tangential to the circle centered at Q constitute the candidate interval for possible horizontal directions. In some cases (left), the sphere point S is too close to the z-axis, which leads to a candidate interval from 0 to 2π.



case 1 *case* 2

Fig. 3. Structural line parameterization.

the geometrical relationship of the horizontal dominant plane and the sphere cap ω decides the possible HDD. Only the HDD whose orthogonal plane intersects the sphere cap ω can produce the line measurement $\vec{O}S$ within a limited noise range. As shown in Fig. 2, this observation leads to a candidate interval $[\beta_1, \beta_2]$ for the associated HDD. Leveraging the MnS algorithm, i.e. the same one also used in the hypothesis step, lines with the common horizontal direction can be identified based on the inlier maximum principle.

The key contribution here lies in the candidate interval computation. The original algorithm directly solves for the angle between the candidate HDD and the x-axis, which leads to cumbersome calculation. Instead, we solve the tangent point T first and then compute the candidate HDD. For the details and derivations of the proposed candidate interval calculation and vanishing point computation method, please refer to the supplementary material.

C. Closed-Form Structural Line Initialization

Before utilizing the structural line features for optimization as shown in (2), the initial value should be determined. Several methods have been proposed for the structural line initialization [25], [31]. However, they cannot either fully exploit the prior information to benefit the initialization or provide a closed-form solution. In this section, we present a closed-form initialization method leveraging the known direction of the structural line.

1) Structural Line Parameterization: Similar to [25], [31], a 2-Dof parameterization is adopted for the structural line. As shown in Fig. 3, l is a structural line in the world frame and π_l is a plane passing through the origin O and orthogonal to l, which is generally named as the projection plane of l. We then use the intersection point P_l of l and π_l to represent the structural line. Since π_l is known given the direction of the structural line, only two extra parameters are needed to parameterize P_l. Fig. 3 depicts two cases for the parameterization. In *case 1* where l is parallel to the z-axis, the x and y coordinates of the intersection point P_l in the world frame are used to parameterize the line. In *case 2* where l is parallel to some Atlanta HDD, a local frame F_l is established and the y and z coordinates of the intersection point P_l in F_l are used for the parameterization of the line.

2) Structure-Aware Closed-Form Initialization: We use multiple-view line observations to obtain an initial estimation for a structural line:

$$\begin{aligned} s_0 n_0 &= R_0^T n_w - R_0^T t_0^{\times} v_w \\ &\vdots \\ s_{M-1} n_{M-1} &= R_{M-1}^T n_w - R_{M-1}^T t_{M-1}^{\times} v_w \end{aligned} \tag{4}$$

where M is the number of views, $\{n_w, v_w\}$ is the Plücker coordinates of the line in the world frame, $\{R, t\}$ is the transformation from the camera frame to the world frame, and $n_i (i = 0, \ldots, M-1)$ is a unit vector representing the line normal in the i-th camera frame. To eliminate the unknown scale s, it follows from (4) that:

$$\begin{aligned} n_0^{\times}(R_0^T n_w - R_0^T t_0^{\times} v_w) &= 0 \\ &\vdots \\ n_{M-1}^{\times}(R_{M-1}^T n_w - R_{M-1}^T t_{M-1}^{\times} v_w) &= 0 \end{aligned} \tag{5}$$

where $(\cdot)^{\times}$ represents the skew symmetric matrix of a vector.

For a structural line, we use the intersection point to parameterize it. The conversion between the intersection point and the Plücker coordinates is as follows:

$$v_w = \begin{cases} (0, 0, 1), & VDD\ line \\ (-\sin(\beta), \cos(\beta), 0), & HDD\ line \end{cases}$$
$$n_w = p_{3d}^{\times}(p_{2\,d}, \beta) \cdot v_w \tag{6}$$

where β is the angle of the HDD and the x-axis, p_{3d} is the coordinate of the intersection point P_l in the world frame which can be calculated from the HDD direction and the 2-Dof strutural line parameterization $p_{2\,d}$ as follows:

$$p_{3d} = \begin{cases} (p_{2d,1}, p_{2d,2}, 0), & VDD\ line \\ (\cos(\beta) \cdot p_{2d,1}, \sin(\beta) \cdot p_{2d,1}, p_{2d,2}), & HDD\ line \end{cases} \tag{7}$$

where $p_{2\,d,i} (i = 1, 2)$ is the i-th element of p_{2d}.

We subsitute (6) and (7) into (5) to formulate a linear equation with respect to the unknown variable $p_{2\,d}$:

$$A_{2M \times 2} \cdot p_{2\,d} = b_{2M} \tag{8}$$

where the SVD method is applied to solve this equation and provide a closed-form solution. Thus the initial value of the structural line feature is obtained as the solution of (8).

V. Experiments

In this section, we present the experiment results of the proposed candidate interval calculation method and the proposed odometry method. For the former, we utilize the open-sourced codeprovided by the authors of [30] to demonstrate the computational efficiency of our method. In particular, the proposed candidate interval calculation method is integrated into the MnS algorithm proposed in [30] for the structural line clustering and the camera rotation estimation. Also, the original candidate

TABLE I
THE ROTATION ERROR [DEGREE] AND THE COMPUTATION TIME [MS] OF TWO METHODS ON THE ICL-NUIM DATASET

methods	lr-kt0	lr-kt1	of-kt0	of-kt1	computation time
[30]	0.24	0.43	0.33	0.22	0.11
ours	0.24	0.43	0.33	0.22	0.01

TABLE II
THE NUMBER OF OPERATIONS FOR A SINGLE CALCULATION

methods	addition	subtraction	multiplication	division	others[1]
[30]	309	306	2955	2	1508
ours	7	8	24	7	7

1: *Others* include exponentiation, square root and trigonometry operations.

TABLE III
ATE [M] OF ALGORITHMS ON THE VCU-RVI DATASET

Sequence	[23]	[2]	[25]	[7]	[8]	s-vio	w/o s.[1]
motion_1	0.41	0.32	**0.15**	0.39	0.29	0.26	0.29
motion_2	lost	0.46	**0.24**	0.37	0.36	0.28	0.38
motion_3	lost	0.26	0.19	0.18	0.17	**0.12**	0.17
motion_4	lost	0.37	**0.23**	0.24	0.35	0.32	0.34
motion_5	lost	0.24	0.27	0.28	**0.14**	0.16	0.22
motion_6	lost	0.28	0.43	0.28	0.27	**0.18**	0.19
dynamic_1	0.50	0.20	0.22	0.20	0.17	**0.10**	0.11
dynamic_2	lost	0.37	0.17	0.18	0.14	**0.08**	**0.08**
dynamic_3	1.65	0.40	**0.09**	0.33	0.27	0.19	0.20
dynamic_4	1.85	1.09	1.06	1.35	1.47	**0.14**	0.16
dynamic_5	lost	0.37	0.12	0.17	0.11	**0.05**	0.24
light_1	lost	1.06	**0.17**	2.62	7.23	1.18	4.39
light_2	lost	0.34	**0.24**	0.55	2.15	0.31	2.08
light_3	lost	0.24	**0.12**	0.24	0.24	**0.12**	0.19
light_4	0.19	0.13	**0.10**	0.16	0.11	**0.10**	0.18
light_5	lost	9.35	**0.31**	8.30	6.47	0.44	13.33
light_6	lost	0.75	**0.43**	0.83	11.45	0.58	4.28
corridor_1	lost	3.81	0.62	1.09	4.66	**0.58**	1.60
corridor_2	lost	**0.79**	1.43	2.55	0.93	1.49	1.94
corridor_3	lost	2.77	**0.68**	7.45	1.68	0.91	0.94
corridor_4	lost	2.72	0.78	2.15	1.98	**0.20**	1.09
hall_1	lost	3.30	1.52	6.18	2.89	**1.21**	3.96
hall_2	lost	2.56	1.89	5.11	1.82	**1.15**	5.68
hall_3	lost	5.34	**1.25**	3.68	3.01	1.51	2.37
Avg. long[2]	omit	3.04	1.17	4.03	2.42	**1.01**	2.51
Avg. short[3]	omit	0.30	0.20	0.25	0.22	**0.16**	0.21

1: S-VIO without the AW detection.
2: The average ATE for the long-range sequences, i.e. *corridor* and *hall*.
3: The average ATE for the short-range sequences, i.e. *motion*, *dynamic* and *light*. It should be noted that sequences where at least one algorithm except [23] produces large positioning error are excluded, including *dynamic_4*, *light_1*, *light_2*, *light_5* and *light_6*.

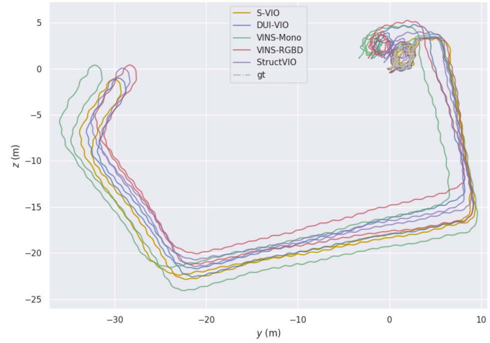


Fig. 4. Qualitative results of evaluated algorithms on *corridor4*. The start and the end of the trajectory should be in the same room.

interval calculation method contained in the open-sourced code is tested to provide a baseline. Both methods are implemented in MATLAB and run on a laptop computer with an Intel Core i7 (2.8 GHz) CPU and 32 GB RAM. For the latter, we evaluate the proposed S-VIO on two real-world RGB-D inertial dataset, VCU-RVI [32] and OpenLORIS-Scene [33]. Since our system is bulit on VINS-Mono [2] and DUI-VIO [8], they are chosen as the baselines. Additionally, a well-known RGB-D inertial algorithm, VINS-RGBD [7], an RGB-D VO leveraging the MW assumption, MSC-VO [23] and a VIO leveraging the AW assumption, StructVIO [25] are also included in the comparison. For all evaluated algorithms, the loop closing module is disabled to only evaluate the odometry performance. All the experiments are performed using the open-sourced code on a desktop with Intel Core i9 CPU (3.6 GHz) and 32 GB RAM.

A. Evaluation of Candidate Interval Calculation

Table I reports the quantitative evaluation results of the two methods on the ICL-NUIM dataset, including the average error of the rotation estimation and the average computation time of the candidate interval calculation per line. Experiment results show that both methods can obtain accurate camera rotation results. Particularly, it is observed that the rotation estimation results of two methods are almost the same, which demonstrates the correctness of our candidate interval calculation method. In terms of the computational speed, the execution time of the proposed candidate interval computation method is an order of magnitude lower than the original method's. This is mainly because we take a different way to establish the equation and thus avoid the cumbersome coefficient calculation in the original method. Table II shows the number of operations needed by a single calculation of the equation coefficients. It is observed that our method requires fewer operations than the original method does. Besides, our method only needs to solve a second-order equation, and thus eliminates the need for high-order equation solving.

B. Evaluation of S-VIO on the VCU-RVI Dataset

The VCU-RVI dataset [32] is an RGB-D inertial dataset recorded in the real-world indoor environment. It contains synchronized images and IMU data from an RGB-D VI sensor suite and captures a large variety of challenging factors, including fast motion, dim illumination, dynamic objects and weak texture. The evo package is used to compute the root mean square error (RMSE) of absolute trajectory error (ATE) with $SE(3)$ Umeyama alignment. For short-range sequences that are collected in one room, i.e. *motion*, *dynamic* and *light*, the whole trajectory is used for the alignment. For long-range sequences that move beyond the room and then return to the start point, i.e. *corridor* and *hall*, the available groundtruth is constrained in the room. In such a case, we use the beginning part of the trajectory until the first time that leaves the room for the alignment, in order to manifest the odometry drift in the long-range operation.

Table III shows the accuracy comparison of evaluated algorithms on the VCU-RVI handheld dataset, including 24 sequences in total. Among all six algorithms, S-VIO achieves

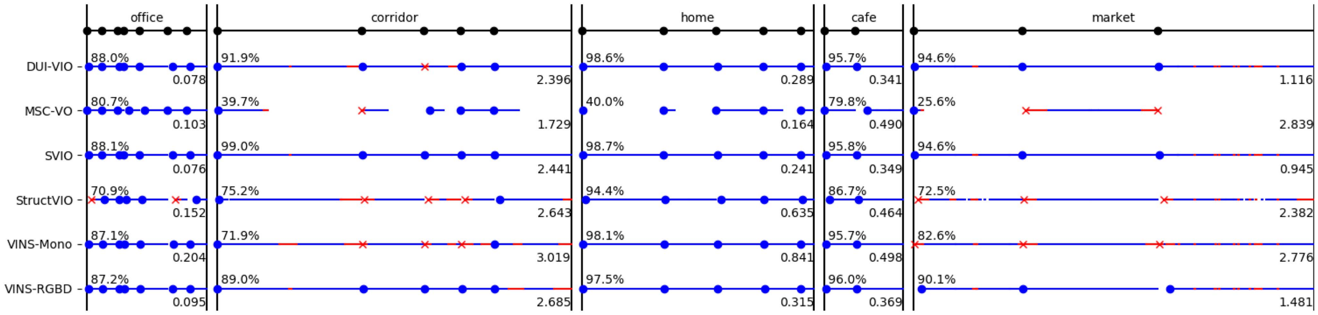


Fig. 5. Evaluation results on the OpenLORIS-Scene dataset. The dataset consists of 22 sequences discontinue in space and time, and each algorithm is tested on them in a sequence-by-sequence fashion. The line belong to each algorithm represents the tracking span which begins with the completion of the initialization and ends with the tracking failure or the ending of the sequence. The average Correct Rate ($CR^{\varepsilon,\phi}$) is labeled on the top left of each scene, larger means more robust, and the average ATE RMSE is labeled on the bottom right, smaller means more accurate. Particularly, the $CR^{\varepsilon,\phi}$ divides the whole tracking span into the correctly tracking part (indicated by the blue line) and the part with large localization error (indicated by the red line).

the lowest or the second-lowest ATE in most sequences. Compared with the algorithms without the structural constraints, in the short-range sequences, S-VIO obtains 25%, 35% and 46% improvement on the average ATE (w/o failures) with respect to DUI-VIO, VINS-RGBD and VINS-Mono. The advantage is more obivous on the long-range sequences, where S-VIO surpasses DUI-VIO, VINS-RGBD and VINS-Mono by a larger margin of 58%, 75% and 67% respectively. This is mainly because the exploitation of the AW assumption reduces the attitude estimation error and thus benefits the accuracy of the odometry. As shown in Fig. 4, in the long term operation, S-VIO has lower localization error compared with the other algorithms. We also perform ablation study to demonstrate the necessity of obtaining structural constraint. The AW detection module in S-VIO is turned off, resulting in absence of the structural line landmarks and the structural constraints on the plane landmarks. As shown in Table III, the accuracy drops compared to S-VIO. Among the structural odometry algorithms, S-VIO obtains a slightly better accuracy than StructVIO does and a considerable improvement over MSC-VO. Specifically, it is observed that compared to StructVIO, S-VIO leveraging the plane feature has a lower drift on the low-textured environment like *corridor4*, where the textureless walls dominate the view for a period of time. The experiment results also illustrate the importance of IMU in terms of the robustness and the accuracy. Although MSC-VO also exploits the structural regularity of the environment in a tightly-coupled fashion, it still suffers from the difficulties caused by the fast motion and the weak texture.

C. Evaluation of S-VIO on the OpenLORIS Dataset

The OpenLORIS-Scene dataset [33] is a real-world indoor dataset containing several challenging factors including weak texture, dynamic/deformable objects and poor illumination. It is collected by two ground robots equipped with multiple sensors. For the evaluation, we only take the data provided by the Realsense D435i camera as the input. Quantitative results on the OpenLORIS-Scene dataset is presented in Fig. 5. For the evaluation metric, the correct rate (CR) defined in [33] is the ratio of the correctly tracked part to the whole time span, and thus can reflect the robustness of the odometry algorithm. The RMSE of ATE is also computed, by only averaging over the correct estimations. All results are computed by the OpenLORIS-Scene-Tools provided by the authors of [33].

TABLE IV
COMPUTATION TIME [MS] OF S-VIO ON THE VCU-RVI DATASET

Feature Tracking Thread	Point Detection & Tracking Line Detection & Tracking Plane Segmentation & Matching Total	13.89 5.83 7.09 14.10
State Estimation Thread	Optimization Marginalization Total	21.45 2.46 28.32

The OpenLORIS-Scene dataset consists of 22 sequences collected in five scenes. In most scenes, the monocular inertial methods, VINS-Mono and StructVIO, perform considerablely worse than the other three RGB-D inertial methods. This is mainly due to the degeneration problem of the monocular VIO on the wheeled robot [34], leading to an inaccurate scale estimate. The pure RGB-D method, MSC-VO easily encounters tracking failure due to the lack of features (i.e. *home* and *corridor*) and the repetitive texture (i.e. *market*), resulting in a low CR. Among the RGB-D inertial methods, VINS-RGBD is less accurate than DUI-VIO and S-VIO. In the *cafe* and *office* scenes that are well-textured and primarily stationary environment, the accuracy of DUI-VIO and S-VIO are comparable. In the *home* and *corridor* scenes, the robot may encounter the diffculties caused by the poor illumination and the weak texture, S-VIO exhibits higher robustness and accuracy than the other algorithms. This is mainly attributed to the popularity of the planar structure in the man-made environment, which can provide the localization constraint even when the robot operates under very dim lighting (i.e. *corridor_3*) or temporarily facing a textureless wall (i.e. *corridor_1* and *home*). In the *market* scene which features dynamic objects, i.e. moving pedestrians and shopping carts, S-VIO also achieves the lowest ATE, by exploiting the line and planar features that are mostly stationary.

D. Runtime Analysis

Table IV reports the runtime breakdown of S-VIO on the VCU-RVI handheld dataset. Both the feature tracking thread and the state optimization thread have an averge runtime below 30 ms, supporting the real-time processing for at least 30 fps image input. It should be noted that the detection and tracking of point, line and plane features are processed in three parallel threads, which saves a lot of computation time.

VI. Conclusion

This letter presents a tightly-coupled RGB-D inertial odometry, namely S-VIO for unknown indoor environment. Heterogeneous landmarks (i.e. points, lines and planes) and multi-modal measurements (i.e. RGB, depth and IMU) are fully utilized to benefit the odometry in the optimization framework. Possible structural regularity (i.e. the Atlanta world) in the environment is also used in this system. Leveraging the gravity direction estimated by VIO, a two-step AW inference method is proposed to detect the horizontal dominant directions from lines and planes. Besides, a closed-form method is proposed to initialize the structural line incorporating the known direction information. Experiments on two public real-world dataset demonstrate that S-VIO achieves better performance compared with state-of-the-art VIO and RGB-D VIO algorithms.

References

[1] C. Campos, R. Elvira, J. J. G. Rodríguez, J. M. M. Montiel, and J. D. Tardós, "ORB-SLAM3: An accurate open-source library for visual, visual–inertial, and multimap SLAM," *IEEE Trans. Robot.*, vol. 37, no. 6, pp. 1874–1890, Dec. 2021.

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

[3] 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.

[4] L. von Stumberg and D. Cremers, "DM-VIO: Delayed marginalization visual-inertial odometry," *IEEE Robot. Automat. Lett.*, vol. 7, no. 2, pp. 1408–1415, Apr. 2022.

[5] H. Li et al., "Globally optimal and efficient vanishing point estimation in atlanta world," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 153–169.

[6] T. Laidlow, M. Bloesch, W. Li, and S. Leutenegger, "Dense RGB-D-inertial SLAM with map deformations," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2017, pp. 6741–6748.

[7] Z. Shan, R. Li, and S. Schwertfeger, "RGBD-inertial trajectory estimation and mapping for ground robots," *Sensors*, vol. 19, no. 10, 2019, Art. no. 2251.

[8] H. Zhang and C. Ye, "DUI-VIO: Depth uncertainty incorporated visual inertial odometry based on an RGB-D camera," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2020, pp. 5002–5008.

[9] J. Liu, X. Li, Y. Liu, and H. Chen, "RGB-D inertial odometry for a resource-restricted robot in dynamic environments," *IEEE Robot. Automat. Lett.*, vol. 7, no. 4, pp. 9573–9580, Oct. 2022.

[10] M. Hsiao, E. Westman, and M. Kaess, "Dense planar-inertial SLAM with structural constraints," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2018, pp. 6521–6528.

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

[12] D. Chen et al., "VIP-SLAM: An efficient tightly-coupled RGB-D visual inertial planar SLAM," in *Proc. Int. Conf. Robot. Automat.*, 2022, pp. 5615–5621.

[13] Y. Lu and D. Song, "Visual navigation using heterogeneous landmarks and unsupervised geometric constraints," *IEEE Trans. Robot.*, vol. 31, no. 3, pp. 736–749, Jun. 2015.

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

[15] X. Li, Y. Li, E. P. Örnek, J. Lin, and F. Tombari, "Co-planar parametrization for stereo-SLAM and visual-inertial odometry," *IEEE Robot. Automat. Lett.*, vol. 5, no. 4, pp. 6972–6979, Oct. 2020.

[16] 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.

[17] L. Zhou, S. Wang, and M. Kaess, "DPLVO: Direct point-line monocular visual odometry," *IEEE Robot. Automat. Lett.*, vol. 6, no. 4, pp. 7113–7120, Oct. 2021.

[18] F. Wu and G. Beltrame, "Direct sparse odometry with planes," *IEEE Robot. Automat. Lett.*, vol. 7, no. 1, pp. 557–564, Jan. 2022.

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

[20] P. Kim, B. Coltin, and H. J. Kim, "Low-drift visual odometry in structured environments by decoupling rotational and translational motion," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2018, pp. 7247–7253.

[21] 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.

[22] Y. Li, R. Yunus, N. Brasch, N. Navab, and F. Tombari, "RGB-D SLAM with structural regularities," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2021, pp. 11581–11587.

[23] J. P. Company-Corcoles, E. Garcia-Fidalgo, and A. Ortiz, "MSC-VO: Exploiting Manhattan and structural constraints for visual odometry," *IEEE Robot. Automat. Lett.*, vol. 7, no. 2, pp. 2803–2810, Apr. 2022.

[24] 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, pp. 203–209.

[25] 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.

[26] 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.*, vol. 44, no. 11, pp. 8403–8419, Nov. 2022.

[27] J. H. Lee, S. Lee, G. Zhang, J. Lim, W. K. Chung, and I. H. Suh, "Outdoor place recognition in urban environments using straight lines," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2014, pp. 5550–5557.

[28] C. Feng, Y. Taguchi, and V. R. Kamat, "Fast plane extraction in organized point clouds using agglomerative hierarchical clustering," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2014, pp. 6218–6225.

[29] L. Zhou, D. Koppel, H. Ju, F. Steinbruecker, and M. Kaess, "An efficient planar bundle adjustment algorithm," in *Proc. IEEE Int. Symp. Mixed Augmented Reality*, 2020, pp. 136–145.

[30] P. Kim, H. Li, and K. Joo, "Quasi-globally optimal and real-time visual compass in Manhattan structured environments," *IEEE Robot. Automat. Lett.*, vol. 7, no. 2, pp. 2613–2620, Apr. 2022.

[31] H. Wei, F. Tang, Z. Xu, and Y. Wu, "Structural regularity aided visual-inertial odometry with novel coordinate alignment and line triangulation," *IEEE Robot. Automat. Lett.*, vol. 7, no. 4, pp. 10613–10620, Oct. 2022.

[32] H. Zhang, L. Jin, and C. Ye, "The VCU-RVI benchmark: Evaluating visual inertial odometry for indoor navigation applications with an RGB-D camera," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2020, pp. 6209–6214.

[33] X. Shi et al., "Are we ready for service robots? the openloris-scene datasets for lifelong SLAM," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2020, pp. 3139–3145.

[34] K. J. Wu, C. X. Guo, G. Georgiou, and S. I. Roumeliotis, "VINS on wheels," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2017, pp. 5155–5162.