2022 IEEE International Conference on Robotics and Automation (ICRA)
May 23-27, 2022. Philadelphia, PA, USA

Integrating Point and Line Features for Visual-Inertial Initialization

Hong Liu, Junyin Qiu*, Weibo Huang

2022 IEEE International Conference on Robotics and Automation (ICRA) | 978-1-7281-9681-7/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICRA46639.2022.9811641

Abstract— **Accurate and robust initialization is crucial in visual-inertial system, which significantly affects the localization accuracy. Most of the existing feature-based initialization methods rely on point features to estimate initial parameters. However, the performance of these methods often decreases in real scene, as point features are unstable and may be discontinuously observed especially in low textured environments. By contrast, line features, providing richer geometrical information than points, are also very common in man-made buildings. Thereby, in this paper, we propose a novel visual-inertial initialization method integrating both point and line features. Specifically, a closed-form method of line features is presented for initialization, which is combined with point-based method to build an integrated linear system. Parameters including initial velocity, gravity, point depth and line's endpoints depth can be jointly solved out. Furthermore, to refine these parameters, a global optimization method is proposed, which consists of two novel nonlinear least squares problems for respective points and lines. Both gravity magnitude and gyroscope bias are considered in refinement. Extensive experimental results on both simulated and public datasets show that integrating point and line features in initialization stage can achieve higher accuracy and better robustness compared with pure point-based methods.**

I. Introduction

Visual-Inertial Odometry (VIO) aims to track the pose of a mobile platform in an unknown environment using a camera and an Inertial Measurement Unit (IMU) as input. In recent years, the fusion sets of visual and inertial sensors attract increasing attention, since they provide complementary information to each other, boosting the performance of motion tracking. Besides, these two types of sensors are small, cheap and low power consumption, therefore it is convenient to equip them on various devices. With the growing demand of applications such as virtual reality, augmented reality and autonomous robot navigation [1]–[3], VIO technique has become an indispensable role in robotic state estimation.

To run a VIO system smoothly, some initial parameters need to be estimated. As for the camera-IMU extrinsic calibration, one can perform it using offline [4], [5] or online [6]–[8] methods for only once as it is time invariant. However, others such as gravity direction, initial velocity, IMU biases and metric scale for monocular case, should be initialized online as they are different every time the system is launched. All of these parameters can be approximately recovered by combining visual and inertial measurements together during a short period. It is a crucial process since bad initial values would provide faulty estimation and corrupt the whole system. Over the last decade, many initialization approaches for VIO system have been proposed, which can be classified into two main categories: state-based and feature-based, depending on how to fuse visual measurements.

In state-based methods, visual measurements are utilized to solve a Structure-from-Motion (SfM) problem independently for state estimation to provide reference camera poses. Meanwhile inertial readings are integrated to get relative motions. Hence, initial parameters can be retrieved via aligning IMU motion with camera poses. This idea was firstly proposed by Mur-Artal and Tardós [9], where parameters were estimated and refined in steps, while longer than 10 seconds were required for convergence. Qin et al. [10] presented a similar method but ignoring accelerometer bias, which shortens the convergence time. Afterwards, Campos et al. [11] formulated the initialization process as a maximum-a-posteriori problem, where all parameters were solved out in one step while holding higher accuracy. Furthermore, Zuñiga-Noël et al. [12] presented an analytical solution for state-based methods, dramatically reducing the solving time. However, all abovementioned methods take visual trajectory as a strict reference consistently, which means the performance of these methods are significantly impacted by the pure visual SfM. Once visual initialization fails or the states estimated by vision are inaccurate, the VIO system will fail to initialize.

By contrast, feature-based methods relate observed features and IMU readings together to solve the problem jointly, avoiding the inconsistencies from decoupled estimation. For example, Martinelli proposed a closed-form method in [13], where the observed points are explicitly fused with IMU preintegration into a linear system without visual initialization. In this way, parameters including initial velocity, gravity and depth of feature points could be recovered simultaneously through finding the linear least squares solution. However, this method is unsuitable for consumer-grade IMU sensors as it ignores the gyroscope bias. Kaiser et al. [14] took the gyroscope bias into account and built a cost function for refinement. Later, this work was extended by Campos et al. [15], who considered the magnitude of gravity and added two rounds of bundle adjustment for further optimization. Recently, Evangelidis and Micusik [16] proposed an elimination strategy to separate the initial parameters from unnecessary point positions thus reducing the solving time. Although the above-mentioned feature-based methods have achieved impressive performance, they all required that several points should be tracked in all frames consistently, therefore their performance was impacted by

This work is supported by National Natural Science Foundation of China (No.62073004), Science and Technology Plan of Shenzhen (No.JCYJ20190808182209321).

The authors are with the Key Laboratory of Machine Perception, Peking University, Shenzhen Graduate School. qjy@stu.pku.edu.cn, {hongliu, weibohuang}@pku.edu.cn

feature tracking. However, point features, which characterize weak geometric properties between frames, are inevitably noised and unstable in situations with large illumination variances, and sometimes disappear when facing low textured environments, resulting in the instability of initialization. For these reasons, more robust features and accurate tracks are desirable in feature-based methods.

Recently, line features have been adopted in VIO system to assist motion tracking [17]–[19], since they are more reliable and resilient in various scenarios. However, most of the line-aided VIO systems still rely on pure point-related state-based or feature-based initialization methods. They ignore the geometrical constraint provided by line features. This situation inspires us to factor line features into initialization process. In fact, lines can be analogically taken into account for visual SfM in state-based methods. However, it is also affected by visual initialization and takes longer time for latter visual bundle adjustment. In contrast, featured-based methods directly fuse camera observations and IMU readings, which is appropriate to integrate line features.

To this end, we first propose a closed-form solution of line features, and relate it to point method to establish an integrated linear system, where initial velocity, gravity, depth of both points and lines' endpoints can be jointly estimated. Then, in the further refinement stage, we construct residuals for each point or line correspondence to build two novel least squares problems, which are more flexible and time saving compared with [14] and [15]. Finally, the combination of point and line constraints is exploited for global optimization. Our contributions are highlighted in threefold:

- A closed-form method of line features is proposed and combined with points to build an integrated linear system for visual-inertial initialization.
- A global nonlinear optimization method is presented to refine the solutions, which consists of two novel least squares problems for points and lines respectively.
- The performance improved by integrating point and line features is showcased through both simulation and real-world experiments.

II. Closed-form Solution

This section details the proposed closed-form solutions for both points and lines without considering IMU biases. Nevertheless, the results of closed-form solution are provided as the initial estimates of further nonlinear optimization.

A. Point Features

Assuming that the camera and extrinsically calibrated (against the camera) IMU are rigidly mounted to a common bracket, therefore the camera motion is strictly related to the IMU. In this way, the relation between camera and IMU (body) coordinate system (CS) can be written as:

$$\mathbf{R}_c^w = \mathbf{R}_b^w \mathbf{R}_c^b \tag{1}$$

$$s \cdot \mathbf{p}_c^w = \mathbf{R}_b^w \mathbf{p}_c^b + \mathbf{p}_b^w \,, \tag{2}$$

where $\{\mathbf{R}_c^b, \mathbf{p}_c^b\}$ is the relative transformation matrix between the camera and IMU that can be calibrated offline. $\{\mathbf{R}_c^w, \mathbf{p}_c^w\}$ and $\{\mathbf{R}_b^w, \mathbf{p}_b^w\}$ are respectively the poses of camera and IMU expressed in the world CS, and $s \in \mathbb{R}^+$ is the scale factor.

Given two consecutive frames at time i and j, and provided the linear velocity $\mathbf{v}_{b_i}^w$ in the world CS at time i, the poses of the IMU are related by:

$$\mathbf{R}_{b_j}^w = \mathbf{R}_{b_i}^w \Delta\tilde{\mathbf{R}}_{ij} \tag{3}$$

$$\mathbf{p}_{b_j}^w = \mathbf{p}_{b_i}^w + \mathbf{v}_{b_i}^w \Delta t_{ij} + \frac{1}{2}\mathbf{g}^w \Delta t_{ij}^2 + \mathbf{R}_{b_i}^w \Delta\tilde{\mathbf{p}}_{ij} \,, \tag{4}$$

where $\mathbf{g}^w$ stands for the gravity vector, and Δt_{ij} denotes the time interval between i and j. The terms of IMU preintegration $\Delta\tilde{\mathbf{R}}_{ij}$ and $\Delta\tilde{\mathbf{p}}_{ij}$ are independent of the current states and gravity. In particular, they can be directly computed by integrating the IMU outputs between two frames [20].

Let us now consider a map point P, observed by a camera in two different frames $\{C_i, C_j\}$. Geometrically transforming the coordinates of P in these two frames to the world CS, we can get equal results:

$$\lambda_i \mathbf{R}_{c_i}^w \mathbf{u}_i + \mathbf{p}_{c_i}^w = \lambda_j \mathbf{R}_{c_j}^w \mathbf{u}_j + \mathbf{p}_{c_j}^w \,, \tag{5}$$

where $\mathbf{u}_i$ is the normalized bearing of point P in frame C_i, and λ_i is the corresponding depth between camera and P. In order to relate IMU measurements to these observations, we rewrite (5) in the body CS using (1) and (2), and then replace the terms related to the world CS by substituting (3) and (4). Finally, it leads to:

$$\begin{aligned}\lambda_i \mathbf{R}_c^b \mathbf{u}_i = \lambda_j \Delta\tilde{\mathbf{R}}_{ij} \mathbf{R}_c^b \mathbf{u}_j + \mathbf{v}_{b_i}^i \Delta t_{ij} + \frac{1}{2}\mathbf{g}^i \Delta t_{ij}^2 \\ + \Delta\tilde{\mathbf{p}}_{ij} + \left(\Delta\tilde{\mathbf{R}}_{ij} - \mathbf{I}_3\right) \mathbf{p}_c^b \,,\end{aligned} \tag{6}$$

where $\mathbf{I}_3$ is the 3×3 identity matrix, and the vectors of linear velocity and gravity are expressed in the i-th frame CS. Note that λ_i and λ_j denote the true depth without scaling as the scale factor s is set to 1, and the integration of acceleration data $\Delta\tilde{\mathbf{p}}_{ij}$ provides absolute scale implicitly.

Equation (6) describes the relation among a pair of observations, IMU measurements, and four unknown quantities $\{\lambda_i, \lambda_j, \mathbf{v}_{b_i}^i, \mathbf{g}^i\}$ which are required for initialization. To extend this relation to multiple pairs and frames, we reconsider m map points $\{P_1, \cdots, P_m\}$, simultaneously observed by n consecutive frames $\{C_1, \cdots, C_n\}$. Without loss of generality, the first frame C_1 is set as reference, which is connected with each subsequent frame for feature matching. In this case, each matching pair leads to an equation, therefore we totally have $3m(n-1)$ rows of equation and $mn+6$ unknowns. By stacking all equations into a linear over-determined system, a compact form is built:

$$\mathbf{\Xi X} = \mathbf{S} \,, \tag{7}$$

where $\mathbf{X} = \left[\mathbf{v}_{b_1}^{1\,\mathrm{T}}, \mathbf{g}^{1\,\mathrm{T}}, \lambda_1^1, \cdots, \lambda_1^m, \cdots, \lambda_n^1, \cdots, \lambda_n^m\right]^\mathrm{T}$ is the unknown vecter. Matrix $\mathbf{\Xi}$ and vector $\mathbf{S}$ can be filled by measurements. Note that the linear system is sparse which can be quickly solved via conjugate gradient.

B. Line Features

Compared with 3-D points, lines provide richer geometric information. There are various forms of line representations with different pros and cons [21]. An effective representation is Plücker coordinate, which over parameterizes a 3D line as a 6-vector $[\mathbf{n}^{\mathrm{T}},\, \mathbf{d}^{\mathrm{T}}]^{\mathrm{T}}$ with $\mathbf{n}^{\mathrm{T}}\mathbf{d} = 0$, where $\mathbf{n}$ is the normal vector of the plane determined by the line and the origin point, and $\mathbf{d}$ is the line direction vector.

Consider a spatial line L with two end-points $\{\mathbf{s},\, \mathbf{e}\}$ observed in frame C_i, the Plücker coordinate of L in C_i can be written as follows:

$$\begin{bmatrix} \mathbf{n}_i \\ \mathbf{d}_i \end{bmatrix} = \begin{bmatrix} \mathbf{s}_i{}^{\wedge}\mathbf{e}_i \\ \mathbf{e}_i - \mathbf{s}_i \end{bmatrix}, \tag{8}$$

where $\mathbf{s}_i$ and $\mathbf{e}_i$ respectively denote corresponding positions in i-th CS, and $(\cdot)^{\wedge}$ is the skew-symmetric matrix of a vector in $\mathbb{R}^3$. Note that the straight lines are infinite, therefore $\{\mathbf{s}_i, \mathbf{e}_i\}$ can be selected arbitrarily on the line without changing its direction. Moreover, the geometry transformation of Plücker coordinates can be defined as:

$$\begin{bmatrix} \mathbf{n}_w \\ \mathbf{d}_w \end{bmatrix} = \begin{bmatrix} \mathbf{R}_{c_i}^{w} & \mathbf{p}_{c_i}^{w\wedge}\mathbf{R}_{c_i}^{w} \\ \mathbf{0} & \mathbf{R}_{c_i}^{w} \end{bmatrix} \begin{bmatrix} \mathbf{n}_i \\ \mathbf{d}_i \end{bmatrix}. \tag{9}$$

Equation (9) shows a simple linear transformation from the i-th CS to the world CS. Now setting C_i as reference frame, then introducing another viewing frame C_j and transforming the same line into the world CS, we can get equal results as (9), therefore:

$$\mathbf{R}_{c_i}^{w}\mathbf{d}_i = \mathbf{R}_{c_j}^{w}\mathbf{d}_j \tag{10}$$

$$\mathbf{R}_{c_i}^{w}\mathbf{n}_i + \mathbf{p}_{c_i}^{w\wedge}\mathbf{R}_{c_i}^{w}\mathbf{d}_i = \mathbf{R}_{c_j}^{w}\mathbf{n}_j + \mathbf{p}_{c_j}^{w\,\wedge}\mathbf{R}_{c_j}^{w}\mathbf{d}_j\,. \tag{11}$$

Further, substituting (1) (3) into (10), and (1)-(4), (10) into (11), after simplification we have:

$$\mathbf{R}_c^b\mathbf{d}_i = \Delta\tilde{\mathbf{R}}_{ij}\mathbf{R}_c^b\mathbf{d}_j \tag{12}$$

$$\begin{aligned} \mathbf{R}_c^b\mathbf{n}_i = \Delta\tilde{\mathbf{R}}_{ij}\mathbf{R}_c^b\mathbf{n}_j - \left(\mathbf{R}_c^b\mathbf{d}_i\right)^{\wedge} \Big(&\mathbf{v}_{b_i}^{i}\Delta t_{ij} \\ &+ \frac{1}{2}\mathbf{g}^i\Delta t_{ij}^2 + \Delta\tilde{\mathbf{p}}_{ij} + \left(\Delta\tilde{\mathbf{R}}_{ij} - \mathbf{I}_3\right)\mathbf{p}_c^b\Big). \end{aligned} \tag{13}$$

Equations (12) and (13) relate spatial lines to IMU integration with $\mathbf{n}_i = \mathbf{s}_i{}^{\wedge}\mathbf{e}_i$ and $\mathbf{d}_i = \mathbf{e}_i - \mathbf{s}_i$. Here $\mathbf{s}_i$ and $\mathbf{e}_i$ are parameterized as unitary bearing vector with unknown scale:

$$\mathbf{s}_i = -\alpha_i\bar{\mathbf{s}}_i, \quad \mathbf{e}_i = \beta_i\bar{\mathbf{e}}_i, \quad \|\bar{\mathbf{s}}_i\| = \|\bar{\mathbf{e}}_i\| = 1, \tag{14}$$

and the j-th frame in the same way. Importing (14) to (12), and dividing by α_i on both sides, we obtain:

$$\mathbf{R}_c^b(\bar{\mathbf{s}}_i + \hat{\beta}_i\bar{\mathbf{e}}_i) = \Delta\tilde{\mathbf{R}}_{ij}\mathbf{R}_c^b(\hat{\alpha}_j\bar{\mathbf{s}}_j + \hat{\beta}_j\bar{\mathbf{e}}_j)\,, \tag{15}$$

with $\hat{\beta}_i = \beta_i/\alpha_i$, $\hat{\alpha}_j = \alpha_j/\alpha_i$ and $\hat{\beta}_j = \beta_j/\alpha_i$. Similarly, as for (13), it can be rewritten as:

$$\begin{aligned} \hat{\lambda}_i\mathbf{R}_c^b\bar{\mathbf{n}}_i = \hat{\lambda}_j\Delta\tilde{\mathbf{R}}_{ij}\mathbf{R}_c^b\bar{\mathbf{n}}_j - \left(\mathbf{R}_c^b\hat{\mathbf{d}}_i\right)^{\wedge} \Big(&\mathbf{v}_{b_i}^{i}\Delta t_{ij} \\ &+ \frac{1}{2}\mathbf{g}^i\Delta t_{ij}^2 + \Delta\tilde{\mathbf{p}}_{ij} + \left(\Delta\tilde{\mathbf{R}}_{ij} - \mathbf{I}_3\right)\mathbf{p}_c^b\Big)\,, \end{aligned} \tag{16}$$

with $\bar{\mathbf{n}}_i = \bar{\mathbf{s}}_i{}^{\wedge}\bar{\mathbf{e}}_i$, $\bar{\mathbf{n}}_j = \bar{\mathbf{s}}_j{}^{\wedge}\bar{\mathbf{e}}_j$, $\hat{\mathbf{d}}_i = \bar{\mathbf{s}}_i + \hat{\beta}_i\bar{\mathbf{e}}_i$ and corresponding scale $\{\hat{\lambda}_i,\, \hat{\lambda}_j\} \in \mathbb{R}$. Since the unitary bearing vector can be directely computed by camera measurements, the unknowns in (15) and (16) are $\{\mathbf{v}_{b_i}^{i},\, \mathbf{g}^i,\, \hat{\beta}_i,\, \hat{\alpha}_j,\, \hat{\beta}_j,\, \hat{\lambda}_i,\, \hat{\lambda}_j\}$. Note that (15) is a simple linear system which can be solved preliminarily, then the obtained $\hat{\mathbf{d}}_i$ can be substituted into (16), formulating another linear system to find remaining unknowns.

To extend to multiple pairs of lines and frames, likewise, let us now reconsider M spatial lines $\{L_1, \cdots, L_M\}$ and N consecutive frames $\{C_1, \cdots, C_N\}$ with the reference frame C_1. We first build M independent linear systems for each line base on (15). The one of them can be written as:

$$\mathbf{A}_i\mathbf{x}_i = \mathbf{b}_i\,, \tag{17}$$

where $\mathbf{x}_i = \left[\hat{\beta}_1^i,\, \hat{\alpha}_2^i,\, \hat{\alpha}_3^i,\, \cdots,\, \hat{\alpha}_N^i,\, \hat{\beta}_2^i,\, \hat{\beta}_3^i,\, \cdots,\, \hat{\beta}_N^i\right]^{\mathrm{T}}$ denotes the unknowns of the i-th line. To recover $\hat{\mathbf{d}}_1^i$ in (16), only $\hat{\beta}_1^i$ is required. In fact, the unknowns except for $\hat{\beta}_1^i$ can be eliminated by left multiplying a certain matrix on both sides of the equation, like the method in [16], since there are two sub-matrices in $\mathbf{A}_i$ which consist of mutually orthogonal unit vectors. In any way, $\hat{\mathbf{d}}_1^i$ can be recovered, thus $\left(\mathbf{R}_c^b\hat{\mathbf{d}}_i\right)^{\wedge}$ in (16) can be regarded as a known 3×3 matrix thereafter.

Let us now turn to (16), by stacking all relations and building a linear over-determined system of observed lines, another compact form is built:

$$\mathbf{\Pi}\mathbf{X} = \mathbf{T}\,, \tag{18}$$

where $\mathbf{X} = \left[\mathbf{v}_{b_1}^{1\;\mathrm{T}}, \mathbf{g}^{1\mathrm{T}}, \hat{\lambda}_1^1, \cdots, \hat{\lambda}_1^M, \cdots, \hat{\lambda}_N^1, \cdots, \hat{\lambda}_N^M\right]^{\mathrm{T}}$ is the unknown vecter. Similarly, for sparsity reason, conjugate gradient is employed to solve (18). Based on the solution, the depths of endpoints can be also recovered if needed.

Furthermore, an integrated closed-form method, consisting of both points and lines, can be formulated expectedly, which combines (7) and (18) with common unknowns $\{\mathbf{v}_{b_1}^1,\, \mathbf{g}^1\}$ and their own parameters.

III. Non-linear Optimization

Note that the IMU biases as well as the magnitude of gravity are not considered until now. Hence, in most case, the solutions above are unreliable. In this section, these considerations are introduced to refine them.

A. Point-based Optimization

In [14] and [15], the parameters are refined by directly minimizing the residual $\|\mathbf{\Xi}\mathbf{X} - \mathbf{S}\|^2$, where the solution $\mathbf{X}$ can be expressed in closed-form using the pseudo-inverse. However, this method ties all relations up and regards each of them equally, causing instability especially when facing some outliers. Moreover, Jacobians are computed numerically for each iteration. When the matrix dimension becomes larger, it's time-consuming.

Instead, it can be noticed that each matching pair of points leads to a relation (6), thus the residual of the i-th point between reference frame and the j-th frame can be defined:

$$\begin{aligned} \mathbf{r}_{\mathbf{p}_{ij}} = \lambda_1^i\mathbf{R}_c^b\mathbf{u}_1^i - \lambda_j^i\Delta\tilde{\mathbf{R}}_{1j}\mathbf{R}_c^b\mathbf{u}_j^i - \mathbf{v}_{b_1}^1\Delta t_{1j} \\ - \frac{1}{2}\mathbf{R}_G^1\mathbf{g}_I\Delta t_{1j}^2 - \Delta\tilde{\mathbf{p}}_{1j} - \left(\Delta\tilde{\mathbf{R}}_{1j} - \mathbf{I}_3\right)\mathbf{p}_c^b\,. \end{aligned} \tag{19}$$

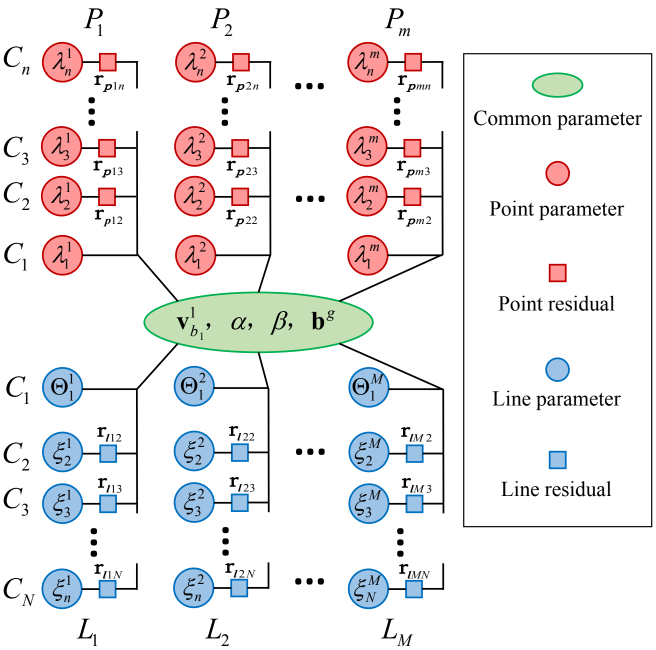

Fig. 1. Graph illustration of points and lines combined optimization problem. The first common frame C_1 is set as reference and is related to each subsequent frame to build residuals. Each residual contains the variables from reference frame, corresponding frame and common parameters.

Here the magnitude of gravity g is fixed by parameterizing $\mathbf{g}^1 = \mathbf{R}_G^1 \mathbf{g}_I$ where $\mathbf{g}_I = (0,\ 0,\ -g)$ and $\mathbf{R}_G^1 = \mathrm{Exp}(\alpha,\ \beta,\ 0)$. Further, we sum all residuals up, taking the gyroscope bias $\mathbf{b}^g$ into account, formulating a non-linear optimization problem:

$$\mathbf{x}_\mathbf{p}^* = \underset{\mathbf{x}_\mathbf{p}}{\mathrm{argmin}} \sum_{i=1}^{m} \sum_{j=2}^{n} \left\| \mathbf{r}_{\mathbf{p}_{ij}} \right\|_{\Sigma_{ij}}^2 , \tag{20}$$

where $\mathbf{x}_\mathbf{p}^* = \{\mathbf{v}_{b_1}^1, \alpha, \beta, \mathbf{b}^g, \lambda_1^1, \cdots, \lambda_1^m, \cdots, \lambda_n^1, \cdots, \lambda_n^m\}$ contains all parameters to be optimized, and their initial values come from closed-form results. Σ_{ij} is the corresponding covariance matrix which is set to identity in this paper. Note that j starts with 2, since the first frame is reference. Similar to [14], the accelerometer bias is not taken into account, as it is coupled with gravity, almost unobservable during a short time [10], and our system is only slightly affected by it.

B. Line-based Optimization

As for lines features, each pair leads to a residual as well, and all parameters are added into relation (16). The constraint of (15) is not considered, since it only provides $\hat{\mathbf{d}}_i$ for (16) but introduces some redundant parameters and complicates the system. Accordingly, in order to optimize $\mathbf{d}_i$ more efficiently, we parameterize it back as $\mathbf{d}_i = \alpha_i \bar{\mathbf{s}}_i + \beta_i \bar{\mathbf{e}}_i$ via dividing by $\hat{\lambda}_i$ on both sides in (16). Hence, the residual for the i-th line between the reference and the j-th frame can be defined as:

$$\begin{aligned}\mathbf{r}_{\mathbf{l}_{ij}} = \mathbf{R}_c^b \bar{\mathbf{n}}_1^i - \xi_j^i \Delta\tilde{\mathbf{R}}_{1j} \mathbf{R}_c^b \bar{\mathbf{n}}_j^i + \left(\mathbf{R}_c^b \mathbf{d}_1^i\right)^\wedge \Big(& \mathbf{v}_{b_1}^1 \Delta t_{1j} \\ + \frac{1}{2} \mathbf{R}_G^1 \mathbf{g}_I \Delta t_{1j}^2 + \Delta\tilde{\mathbf{p}}_{1j} + \left(\Delta\tilde{\mathbf{R}}_{1j} - \mathbf{I}_3\right) \mathbf{p}_c^b & \Big) ,\end{aligned} \tag{21}$$

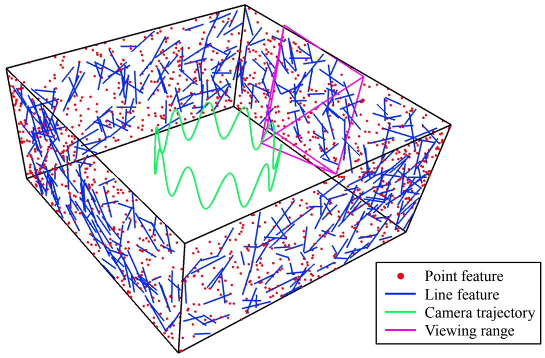

Fig. 2. Simulation setting: The camera moves along the circular trajectory with sinusoidal motion, observing points and lines randomly generated on the walls of a square environment. Here lines of different length are depicted with their two end-points.

where $\xi_j^i = \hat{\lambda}_j^i / \hat{\lambda}_1^i$, $\mathbf{d}_1^i$ denotes the direction of the i-th line in the reference frame, containing two unknowns $\{\alpha_1^i,\ \beta_1^i\}$. Naturally, the non-linear optimization problem for lines can be obtained by summing all residuals up:

$$\mathbf{x}_\mathbf{l}^* = \underset{\mathbf{x}_\mathbf{l}}{\mathrm{argmin}} \sum_{i=1}^{M} \sum_{j=2}^{N} \left\| \mathbf{r}_{\mathbf{l}_{ij}} \right\|_{\Sigma_{ij}}^2 . \tag{22}$$

$\mathbf{x}_\mathbf{l}^* = \{\mathbf{v}_{b_1}^1, \alpha, \beta, \mathbf{b}^g, \Theta_1^1, \cdots, \Theta_1^M, \xi_2^1, \cdots, \xi_2^M, \cdots, \xi_N^1, \cdots, \xi_N^M\}$. Here $\Theta_1^i = \{\alpha_1^i,\ \beta_1^i\}$ indicates the direction of $\mathbf{d}_1^i$.

C. Global Optimization

In order to realize a more robust and accurate system, it is neccessary to utilize both point and line features. Therefore, we set the common first frame as reference between points and lines, then combining (20) and (22), building a global optimization problem:

$$\mathbf{x}^* = \underset{\mathbf{x}}{\mathrm{argmin}} \left(\sum_{i=1}^{m} \sum_{j=2}^{n} \left\| \mathbf{r}_{\mathbf{p}_{ij}} \right\|_{\Sigma_{ij}}^2 + \sum_{i=1}^{M} \sum_{j=2}^{N} \left\| \mathbf{r}_{\mathbf{l}_{ij}} \right\|_{\Sigma_{ij}}^2 \right) , \tag{23}$$

with $\mathbf{x}^* = \mathbf{x}_\mathbf{p}^* \bigcup \mathbf{x}_\mathbf{l}^*$, as graph illustration shown in Fig. 1.

Equation (23) relates the observed point features, line features and IMU measurements, formulating as a global optimization problem. The Levenberg-Marquardt algorithm is implemented to solve it, whereby the Jacobian of each least squares problem can be computed analytically.

IV. Experiments and Discussions

To evaluate the performance of the proposed closed-form solution and optimization methods, both simulations and real-world experiments are carried out in an Intel Core i7-9700 computer with 16 GB of RAM.

A. Simulation Experiments

Similar to the experimental setting of [20], we simulate a camera executing a circular trajectory of three meter radius with a sinusoidal vertical motion. While moving, the camera, with the frame rate of 10 Hz, observes randomly generated points and lines on the walls of a square environment, as depicted in Fig. 2. The simulated IMU poses are obtained

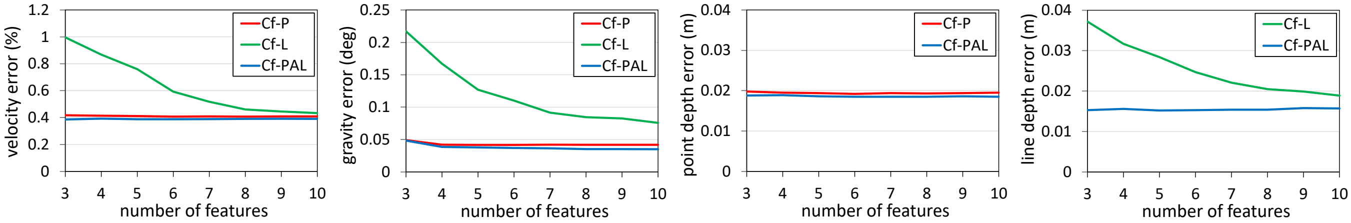

Fig. 3. Comparison of the closed-form solutions with points(Cf-P), lines(Cf-L) and points aided by lines(Cf-PAL) in the case of different tracking features.

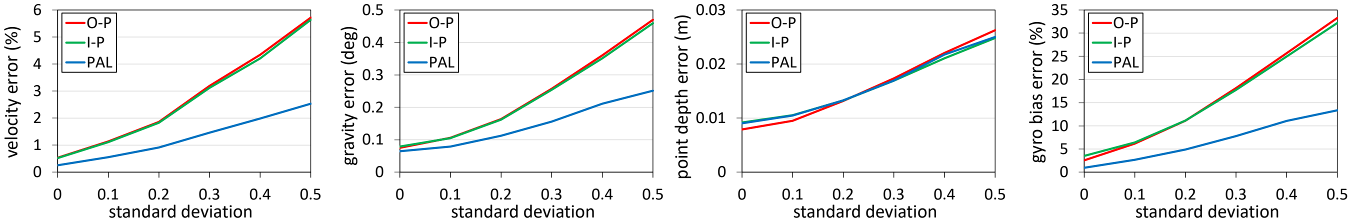

Fig. 4. Errors of initial velocity, gravity direction, gyroscope bias and point depth, with respect to feature noise standard deviation σ in pixel.

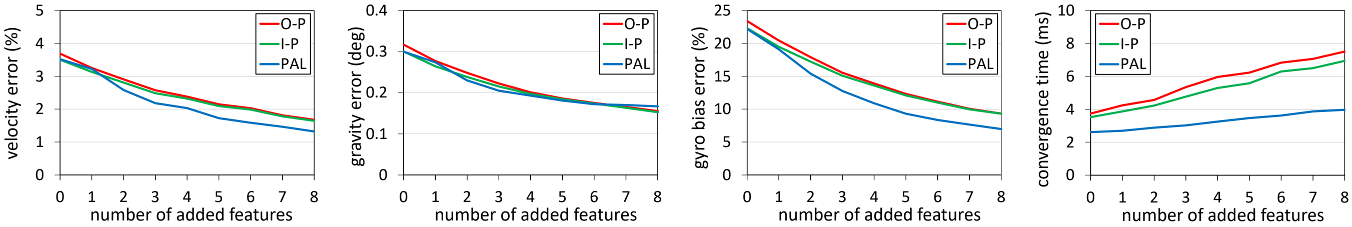

Fig. 5. Effect of adding different corresponding features (points for O-P and I-P, lines for PAL) on the basis of 8 points. Here the comparison of convergence time is presented instead of point depth error.

by transforming using the camera-IMU extrinsic parameters, and its outputs are generated at 200 Hz by computing the analytical derivatives of the given trajectory, while disturbed by Gaussian noise.

We firstly validate the performance of three closed-form solutions: points (Cf-P), lines (Cf-L) and points aided by lines (Cf-PAL), in the cases of different tracking features. Here the IMU biases and observation noise are not considered. The duration time for initialization is 1 second, with 11 frames and 200 integration intervals. For evaluation, the magnitude of initial velocity, gravity direction, point depth and line's endpoints depth with respect to their normalized bearing vector are totally recorded. For generality, all results are generated by averaging ten tests, where each test lasts 20 seconds, for camera moves one lap. The initialization procedure is attempted every 0.1 second.

As shown in Fig. 3, the increase of features has little effect on Cf-P and Cf-PAL, while promotes the accuracy of Cf-L. Although these three closed-form methods are able to estimate initial parameters, the pure line-based method, Cf-L, is inferior to others. This may be caused by its two steps of solving, which are consistency lost. Fortunately, it can be observed carefully that Cf-PAL slightly outperforms Cf-P, preliminarily confirming the effectiveness of line's aiding.

For further validation, both gyroscope bias and observation noise are taken into account to test the refinement method. Gyroscope bias is added manually in simulation, with $\mathbf{b}^g = [-0.0023,\ 0.0249,\ 0.0817]$, similar to the values of EuRoC dataset [22]. In this case, the performance in terms of observation noise deviation σ, ranging from 0 to 0.5 in pixel, is evaluated. We highlight that as for a line feature, noise deviation is added into both of its two endpoints. Additionally, the duration time turns to 2 seconds, while 10 points and 6 lines are tracked over a duration of integration. For comparison, we test the original point-based method (O-P) in [14], the improved form (I-P) in [15] which takes gravity magnitude into consider, and the proposed optimization method combining points and lines (PAL). The results in Fig. 4 show that the proposed PAL outperforms the other two pure point-based methods in terms of estimating velocity, gravity and gyroscope bias. In particular, the errors of velocity and gyroscope bias are reduced almost by half with line features aid compared with O-P and I-P. For point depth, O-P performs better when $\sigma < 0.2$ but is gradually surpassed by I-P and PAL as σ increases, indicating a slight improvement in noised case by considering gravity magnitude.

Besides, to clarify the effectiveness of lines compared with points based on the same feature quantity, another simulation is conducted. We set 8 points as common features for the three methods, while adding the same quantity of extra respective features, e.g., points for O-P and I-P, lines for PAL. Fig. 5 shows the results with respect to the number of added features from 0 to 8. It can be noticed that with the addition

TABLE I
ERRORS OF VELOCITY AND GRAVITY OVER FIVE INITIALIZATION METHODS ON MACHINE HALL SEQUENCES OF EUROC

		Cf-P	Cf-PAL	O-P	I-P	PAL
MH01	vel.(m/s)	0.219	0.220	0.152	0.140	**0.120**
	grav.(deg)	4.61	4.60	1.72	**1.48**	1.50
MH02	vel.(m/s)	0.282	0.282	0.219	0.190	**0.161**
	grav.(deg)	4.66	4.67	1.91	1.65	**1.41**
MH03	vel.(m/s)	0.370	0.371	0.305	0.203	**0.144**
	grav.(deg)	4.70	4.69	2.64	2.09	**1.67**
MH04	vel.(m/s)	0.475	0.477	0.382	0.310	**0.306**
	grav.(deg)	4.70	4.69	2.67	2.18	**1.85**
MH05	vel.(m/s)	0.587	0.586	0.470	0.415	**0.333**
	grav.(deg)	4.99	4.98	2.49	2.24	**1.63**

of features, line is superior to point in terms of estimating velocity and gyroscope bias. Note that the point depth error is not recorded, since PAL and the other two methods track different quantity of points. The comparison of convergence time is also presented in Fig. 5. The results show that with the number of features increases, the convergence time of PAL is shorter and rises more gently than O-P and I-P, implying that the proposed optimization strategy is able to shorten the solving time to a certain extent.

B. Real-World Experiments

Real-world experiments are carried out in EuRoC dataset, which contains synchronized IMU readings with stereo images, and provides the ground-truth of velocities, trajectories and IMU biases. In this work, we only adopt the left camera and the IMU sensors to conduct our system. For point features, we uniformly detect at least 120 corners [23] per frame and use Lucas-Kanade algorithm [24] to track them. The RANSAC with a fundamental matrix model is also adopted for inliers identification. For line features, we adopt FLD algorithm [25] available in OpenCV to detect line segments and use LBD [26] to describe them for feature matching. Lines with their length shorter than 50 in pixel are rejected.

We evaluate the closed-form methods Cf-P, Cf-PAL and optimization methods O-P, I-P, PAL, using machine hall dataset from EuRoC, which consists of five sequences with different complexity. To maintain the consistency of feature quantity, the same number of features are utilized, i.e., 15 points for pure point based methods, while 10 points and 5 lines for line aided methods. With the setting of 2 seconds duration for integration and 10Hz frame updating, those methods are performed once every half second. Errors of the common parameters including absolute initial velocity and gravity direction are presented in Table I. It shows that the performance of the two closed-form solutions (Cf-P, Cf-PAL) are nearly the same, both be surpassed by optimization methods expectedly. Compared with O-P and I-P, PAL achieves more accurate results in most cases, indicating that the addition of line features is able to boost the performance of pure point based methods.

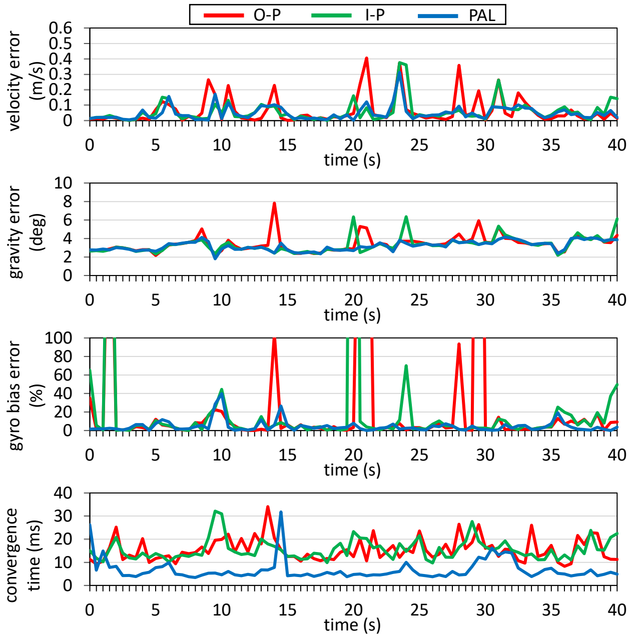

Fig. 6. Results of velocity, gravity, gyroscope bias, and convergence time on V101 sequence, where the first 40 seconds are presented. Feature settings: 15 points for O-P and I-P, while 10 points aided by 5 lines for PAL.

With the same experimental settings, another experiment is carried out in V101 sequence of EuRoC to verify the robustness of lines. Three optimization methods above are employed without limiting their convergence time, and the results over the first 40 seconds are selected for visualization, as seen in Fig. 6. The spikes in each chart indicate bad solutions, which may caused by ineluctable feature noise or tracking error, giving a weak observability of these parameters. The results show that the proposed PAL method effectively avoids some potential spikes in O-P and I-P. Note that the estimation of gravity direction is coupled with accelerometer bias, thus a certain gap between recorded error and zero is formed naturally. Besides, the average convergence time of PAL is mostly less than 10 milliseconds in spite of some fluctuations, while the original approaches in O-P and I-P are both over 10 milliseconds even longer.

V. CONCLUSIONS

In this paper, we propose a novel visual-inertial initialization method by integrating both point and line features. Based on the solution for points, an analogical closed-form solution of line features is proposed, which is related to points to build an integrated linear system to be solved. In addition, two novel nonlinear least squares systems for point and line features are also introduced and suggested to formulate a global optimization problem to refine the closed-form solutions. To verify the effectiveness of the proposed method, both simulations and real-world experiments are conducted. Extensive results show that line features are in possession of better geometrical properties and more robust compared with points. The performance of feature-based method can be improved by integrating point and line features. In the future work, we will investigate the convergence criteria for discarding the solutions with large deviations.

References

[1] W. Fang, L. Zheng, H. Deng, and H. Zhang, “Real-time motion tracking for mobile augmented/virtual reality using adaptive visual-inertial fusion,” *IEEE Sensors Journal*, vol. 17, no. 5, p. 1037, 2017.

[2] P. Li, T. Qin, B. Hu, F. Zhu, and S. Shen, “Monocular visual-inertial state estimation for mobile augmented reality,” in *IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, 2017, pp. 11–21.

[3] Z. Yang, F. Gao, and S. Shen, “Real-time monocular dense mapping on aerial robots using visual-inertial fusion,” in *IEEE International Conference on Robotics and Automation (ICRA)*, 2017, pp. 4552–4559.

[4] J. Rehder and R. Siegwart, “Camera/IMU calibration revisited,” *IEEE Sensors Journal*, vol. 17, no. 11, pp. 3257–3268, 2017.

[5] P. Furgale, J. Rehder, and R. Siegwart, “Unified temporal and spatial calibration for multi-sensor systems,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2013, pp. 1280–1286.

[6] W. Huang and H. Liu, “Online initialization and automatic camera-IMU extrinsic calibration for monocular visual-inertial SLAM,” in *IEEE International Conference on Robotics and Automation (ICRA)*, 2018, pp. 5182–5189.

[7] W. Huang, H. Liu, and W. Wan, “An online initialization and self-calibration method for stereo visual-inertial odometry,” *IEEE Transactions on Robotics*, vol. 36, no. 4, pp. 1153–1170, 2020.

[8] W. Huang, W. Wan, and H. Liu, “Optimization-based online initialization and calibration of monocular visual-inertial odometry considering spatial-temporal constraints,” *IEEE Sensors Journal*, vol. 21, no. 8, p. 2673, 2021.

[9] R. Mur-Artal and J. D. Tardós, “Visual-inertial monocular SLAM with map reuse,” *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 796–803, 2017.

[10] T. Qin and S. Shen, “Robust initialization of monocular visual-inertial estimation on aerial robots,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017, pp. 4225–4232.

[11] C. Campos, J. M. M. Montiel, and J. D. Tardós, “Inertial-only optimization for visual-inertial initialization,” in *IEEE International Conference on Robotics and Automation (ICRA)*, 2020, pp. 51–57.

[12] D. Zuñiga-Noël, F. Moreno, and J. Gonzalez-Jimenez, “An analytical solution to the IMU initialization problem for visual-inertial systems,” *IEEE Robotics and Automation Letters*, 2021.

[13] A. Martinelli, “Closed-form solution of visual-inertial structure from motion,” *International Journal of Computer Vision*, vol. 106, no. 2, pp. 138–152, 2014.

[14] J. Kaiser, A. Martinelli, F. Fontana, and D. Scaramuzza, “Simultaneous state initialization and gyroscope bias calibration in visual inertial aided navigation,” *IEEE Robotics and Automation Letters*, vol. 2, no. 1, pp. 18–25, 2017.

[15] C. Campos, J. M. M. Montiel, and J. D. Tardós, “Fast and robust initialization for visual-inertial SLAM,” in *IEEE International Conference on Robotics and Automation (ICRA)*, 2019, pp. 1288–1294.

[16] G. Evangelidis and B. Micusik, “Revisiting visual-inertial structure-from-motion for odometry and SLAM initialization,” *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 1415–1422, 2021.

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

[18] R. Gomez-Ojeda, F. 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 Transactions on Robotics*, vol. 35, no. 3, pp. 734–746, 2019.

[19] 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,” *arXiv e-prints*, pp. arXiv–2009, 2020.

[20] C. Forster, L. Carlone, F. Dellaert, and D. Scaramuzza, “On-manifold preintegration for real-time visual–inertial odometry,” *IEEE Transactions on Robotics*, vol. 33, no. 1, pp. 1–21, 2017.

[21] A. Bartoli and P. Sturm, “Structure-from-motion using lines: Representation, triangulation, and bundle adjustment,” *Computer vision and image understanding*, vol. 100, no. 3, pp. 416–441, 2005.

[22] M. Burri, J. Nikolic, P. Gohl, T. Schneider, J. Rehder, S. Omari, M. W. Achtelik, and R. Siegwart, “The EuRoC micro aerial vehicle datasets,” *International Journal of Robotics Research*, vol. 35, no. 10, pp. 1157–1163, 2016.

[23] J. Shi and C. Tomasi, “Good features to track,” in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 1994, pp. 593–600.

[24] B. D. Lucas and T. Kanade, “An iterative image registration technique with an application to stereo vision,” in *Proc. Int. Joint Conf. Artif. Intell.*, 1981, pp. 24–28.

[25] 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 *IEEE International Conference on Robotics and Automation (ICRA)*, 2014, pp. 5550–5557.

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