Journal of Intelligent & Robotic Systems (2023) 108:40
https://doi.org/10.1007/s10846-023-01886-3

REGULAR PAPER

A ZUPT Aided Initialization Procedure for Tightly-coupled Lidar Inertial Odometry based SLAM System

Linqiu Gui[1] · Chunnian Zeng[2] · Samuel Dauchert[3] · Jie Luo[2] · Xiaofeng Wang[3]

Received: 13 July 2022 / Accepted: 10 May 2023 / Published online: 24 June 2023

Abstract
Simultaneous localization and mapping (SLAM) is an important research topic in unmanned platforms. The SLAM method, coupled with the Inertial Measurement Unit (IMU), has recently received much attention due to its excellent performance. However, the tightly coupled LiDAR and IMU odometry (LIO) will be interfered with by the unknown initial state during system startup, which will lead to system startup failure in severe cases, especially in the case of low-cost sensors. Therefore, this paper presents a Zero Velocity Update (ZUPT) aided initialization procedure to estimate unknown initial states assuming that the vehicle always starts from stationary. The proposed initialization procedure includes two phases: static phase and dynamic phase, and this paper also proposes a Zero Velocity Detector to distinguish the two phases precisely. The proposed system is evaluated on the datasets gathered from the campus zone with low-cost sensors to evaluate the whole system. The results show that our approach can effectively improve robustness and partly improve precision when the system is bootstrapped.

Keywords LiDAR SLAM · Unmanned platform · Initialization procedure · Zero Velocity Update (ZUPT)

1 Introduction

In the past decade, impressive progress on SLAM has been made [1]. Using a camera or LiDAR, integrated with other devices such as Inertial Measurement Unit (IMU) and GPS, SLAM methods can provide excellent 6-DOF state estimation and 3D world mapping [2]. Furthermore, among these approaches, LiDAR SLAM methods show outstanding robustness to external factors such as illumination, texture, and range, making LiDAR methodologies more suitable than visual SLAM in some instances [3]. Some excellent works such as LOAM [4], Lego-LOAM [5], Segmap [6] and MULLS [7] are all made contributions to the LiDAR SLAM.

A LiDAR SLAM framework usually includes four parts: front-end odometry, back-end optimization, loop closure, and mapping. This paper mainly focuses on the front-end approaches, especially the tightly-coupled LiDAR inertial odometry (LIO) method.

The LIO method has two significant improvements compared with the LiDAR odometry approach. First, the LiDAR odometry method utilizes scan-matching to obtain the relative transformation between two consecutive frames. Scan-matching requires solving a nonlinear optimization problem that heavily relies on a good initial guess. The preintegration of IMU measurements can provide a more accurate initial guess for scan-matching at each computation cycle, thanks to IMU's short-term accuracy. Second, in the LIO method, the LiDAR odometry and the IMU measurements will be jointly optimized to estimate the states, including IMU bias. As a result, the LIO method can outperform LiDAR-only odometry under most conditions [8].

✉ Jie Luo
luo_jie@whut.edu.cn

Linqiu Gui
guilinqiu@whut.edu.cn

Chunnian Zeng
ZengChn@whut.edu.cn

Samuel Dauchert
dauchert@email.sc.edu

Xiaofeng Wang
WANGXI@cec.sc.edu

1 School of Information Engineering, Wuhan University of Technology, Wuhan, Hubei, People's Republic of China

2 School of Automation, Wuhan University of Technology, Wuhan, Hubei, People's Republic of China

3 Department of Electrical Engineering, University of South Carolina, Columbia, SC, USA

However, the LIO method will suffer from the unknown initial states when the system startup, mainly including IMU bias and posture. The IMU preintegration value with unknown initial states is imprecise. Utilizing it as the initial guess for scan matching will quickly make it trapped into a local minimum. In addition, the conjunction optimization of LiDAR odometry and the IMU measurements will also cause the error. The accumulated error may cause system divergence as the system iterates, especially in some application scenarios with low-cost sensors.

Therefore, it is necessary for the system to estimate the unknown initial state at the beginning. However, little recent research has addressed this aspect. In light of this problem, this work proposes a novel initialization procedure for the LIO method that consists of two phases - dynamic and static - which are jointly optimized to estimate the initial states. The static phase utilizes the Zero Velocity Update (ZUPT) method, while the dynamic phase aligns the IMU preintegration's value with the LiDAR odometry when the platform is in motion. The transition point between the two phases is crucial for the initialization procedure's performance, so we propose a Zero Velocity Detector with high robustness to distinguish between the two phases.

Compared with the initialization method that only aligns the pre-integrated IMU odometry with the LiDAR odometry, this paper adds zero velocity updates (ZUPT) to provide more constraints and integrates them into an optimization process, improving the accuracy of state estimation. While many ZUPT-aided inertial navigation methods use the IMU to achieve Zero Velocity Detection, the IMU is easily affected by vibration interference. To enhance the robustness of Zero Velocity Detection, this paper proposes a Zero Velocity Detector that combines the inertial-based detection method with the LiDAR-based detection method. This approach helps to overcome the limitations of the IMU-only method and also enhances the LiDAR-only approach by increasing its ability to resist dynamic object interference, thus improving the overall performance of the initialization procedure.

The main contributions of our work are as follows:

- We propose a ZUPT-aided initialization procedure to estimate the unknown initial states, which improves the robustness of the LIO-based SLAM system when it is bootstrapped.
- We propose a Zero Velocity Detector to detect the stationary for the initialization procedure. The detector combines an inertial-based detection method with a LiDAR-based detection method and achieves high precision and anti-interference properties.

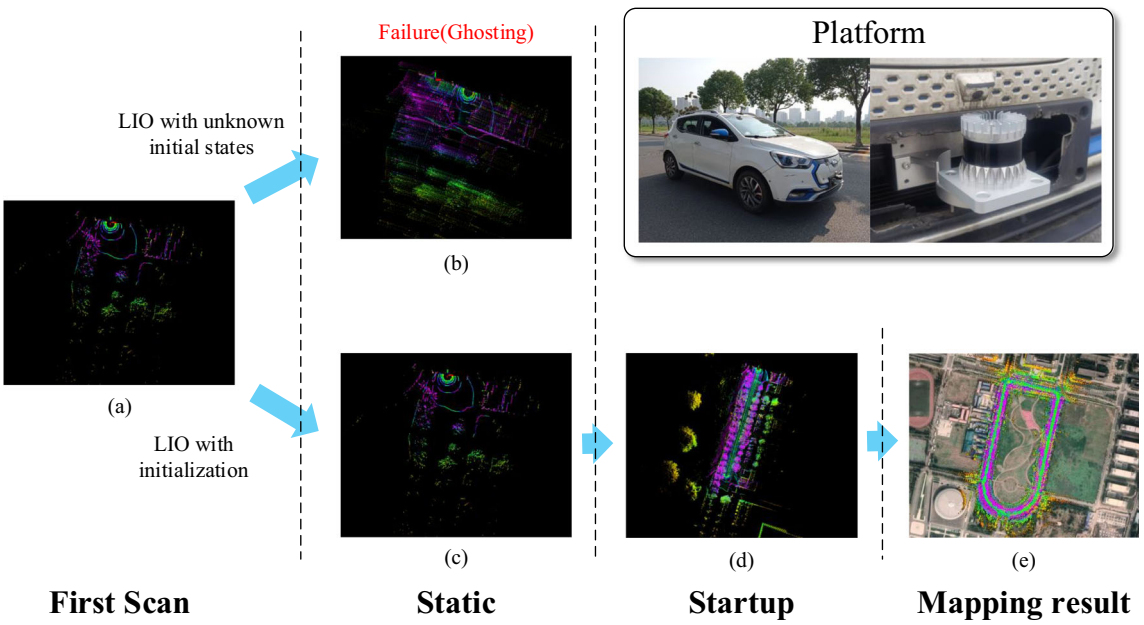


Fig. 1 The Brief Result. We used electric vehicles as a platform in our experiment. The sensor suite includes ouster OS1-16 LiDAR and vigor technology TT810 IMU. We install the LiDAR in front of the vehicle and install the IMU inside the vehicle.4 reference btween IMU and LIDAR was calibrated in advance. Figure **a** illustrates the first LiDAR scan, and the point cloud is sparse. Figure **b** illustrates the LIO result with unknown initial states, and the ghosts show that the system diverges even when the system is static. As shown in figures **c** and **d**, benefiting from the initialization procedure, our method can successfully start from this extreme environment. Figure **e** illustrates an established map aligned with Google Earth, showing the system can establish a points cloud map with good precision. So, our work can effectively solve this problem

- We gather the datasets from the campus zone with low-cost sensors to evaluate the proposed system. The system obtained promising performance, and the brief result is shown in Fig. 1.

The rest of the paper is organized as follows. First, Section 2 discusses the related work. Then, Section 3 presents an overview of the system framework. Next, we discuss the LiDAR odometry and IMU preintegration method in Section 4, which is the base of the initialization procedure and LIO. Then we discuss the LIO method in Section 5 and demonstrate the necessity of the initialization procedure. The initialization procedure is presented in Section 6. Moreover, the global optimization can be found in Section 7. The experimental results are presented in Section 8.

2 Related Work

This section briefly discusses the prior work related to LiDAR inertial odometry, System Initialization and Zero Velocity Update and Zero Velocity Detection.

1. *LiDAR inertial odometry*: The LiDAR inertial odometry (LIO) is a critical component of a LiDAR SLAM system that combines LIDAR odometry with IMU integration to estimate system pose. Where LIDAR odometry uses scan-matching methods such as ICP [9] and GICP [10], as well as feature-based scan matching methods like LOAM [4] and MULLS [7]. Feature-based scan matching methods are more popular due to their computational efficiency.
   LIO can be classified into loosely-coupled and tightly-coupled methods. The famous work Lego-LOAM [5] is a loosely-coupled method that uses IMU to de-skew the LiDAR scan and provide an initial guess for scan-matching. Previous work [11, 12] fused LiDAR with IMU using extended Kalman filters (EKF). Recently, tightly-coupled LIO has attracted broad attention due to the IMU preintegration method [13], which significantly improves its performance. R-LINS [14] uses an error-state Kalman filter in a tightly-coupled manner to estimate the state, and [8] jointly minimizes the cost derived from LiDAR and IMU measurements to achieve better accuracy. LIO-SAM [3] is based on the LOAM framework and utilizes a factor graph to model the LIO. It uses an incremental smoothing method to solve it, which achieves excellent performance and significantly decreases the runtime.
   It's important to note that the LIO method is sensitive to the initial state value, particularly when using low-cost sensors. Therefore, an efficient initialization procedure is essential for improving the robustness of LIO systems during startup.
2. *System Initialization*: SLAM systems tightly coupled with IMU are highly dependent on the initial posture and IMU bias during system startup, as they have a significant impact on the system's robustness and accuracy. As for visual-SLAM, an accurate initialization procedure is crucial, especially when using a monocular camera [15, 16] due to the metric scale. The VINS algorithm [17] provides an efficient initialization procedure by aligning visual odometry with IMU integration odometry for optimization, which results in the necessary values for system startup, including metric scale, posture, IMU bias, and three-dimensional feature location.
   In contrast, there is little research on the initialization procedure for LiDAR SLAM. Currently, the initial posture is usually estimated using the posture output from a 9-axis IMU or the average information from a 6-axis IMU when the system is stationary. The use of LIDAR odometry aligned with IMU integration to estimate the initial posture and IMU bias has also been mentioned [8] but requires further investigation. However, practical application work has shown that the LIO method is also sensitive to the initial state value, especially when using low-cost sensors. Therefore, we propose an initialization procedure for the LIO system to improve robustness during startup.
3. *Zero Velocity Update and Zero Velocity Detection*: ZUPT is a cheap and effective method to solve the long-time accuracy of the inertial navigation system. It utilized the Zero Velocity output as the observation of the velocity error to correct other errors. ZUPT was mainly applied in foot-mounted inertial navigation (motion-tracking) systems [18, 19] and unmanned vehicle applications [20, 21]. Applying the ZUPT methods needs an accurate detection of the stopping time intervals. There are many ways to achieve Zero Velocity Detection. Due to cost and precision concerns, the IMU measurements are mainly used for Zero Velocity Detection. Paper [22] constructs a Markov model to use gyroscope outputs segmentation for detection, making zero velocity detection more reliable. Paper [23] presents the performance of four zero-velocity detectors for a foot-mounted inertial sensor-based pedestrian navigation system. Paper [24] evaluates several commonly used IMU-based Zero Velocity Detection methods. In [25], a frequency domain approach is proposed, using only IMU data to detect stationary, with specifications and analysis for land vehicles. The method in [21] combines sophisticated deep learning techniques with state-of-the-art filtering methods for pure inertial navigation on wheeled vehicles.

Table 1 Mathematical and physical symbols

Symbols	Name
$SO(3)$	Three-Dimensional Special Orthogonal Group
$SE(3)$	Three-Dimensional Special Euclidean Group
$\mathfrak{so}(3)$	Lie Algebra of $SO(3)$
$\mathfrak{se}(3)$	Lie Algebra of $SE(3)$
$\doteq$	Approximate Equal
$\|*\|$	Mahalanobis Distance
$\mathbb{F}$	LiDAR Frame (Message)
X	Platform State
$a \in \mathbb{R}^3$	Acceleration
$w \in \mathbb{R}^3$	Angular Velocity
$b^a \in \mathbb{R}^3$	Acceleration Bias
$b^w \in \mathbb{R}^3$	Gyroscope Bias
$b = \left[b^{a\top}, b^{\omega\top}\right]^\top$	IMU Bias
$p \in \mathbb{R}^3$	Position Vector
$v \in \mathbb{R}^3$	Velocity Vector
$R \in SO(3)$	Rotation Matrix
$T \in SE(3)$	Transformation Matrix
$\phi \in \mathfrak{so}(3)$	Elements of the Lie Algebra Corresponding to R
$\xi \in \mathfrak{se}(3)$	Elements of the Lie Algebra Corresponding to T
W	World Frame
O	Odometry Frame
B	Platform Body Frame (IMU Frame)
G	GPS Frame
L	LiDAR Frame (Coordinate)

3 System Overview

3.1 Definition and Notation

The mathematical and physical symbols used in this paper are defined in Table 1, and we give a supplementary explanation to some symbols as follows: We assume the IMU frame coincides with the platform body frame **B** for convenience; The odometry frame **O** is defined as horizontal to world frame **W**, and we assume the platform's starting heading and position as the odometry frame **O** 's x-axis and origin, respectively; The platform's state X can be defined as:

$$X = [R^\top, p^\top, v^\top, b^\top]^\top \tag{1}$$

In addition, we utilize the symbol $(\cdot)^\wedge$ to define a mapping form Lie Algebra to Lie Group and the symbol $(\cdot)^\vee$ define the reverse mapping:

$$\begin{aligned} \exp(\phi^\wedge) = R, \log(R)^\vee = \phi \\ \exp(\xi^\wedge) = T, \log(T)^\vee = \xi \end{aligned} \tag{2}$$

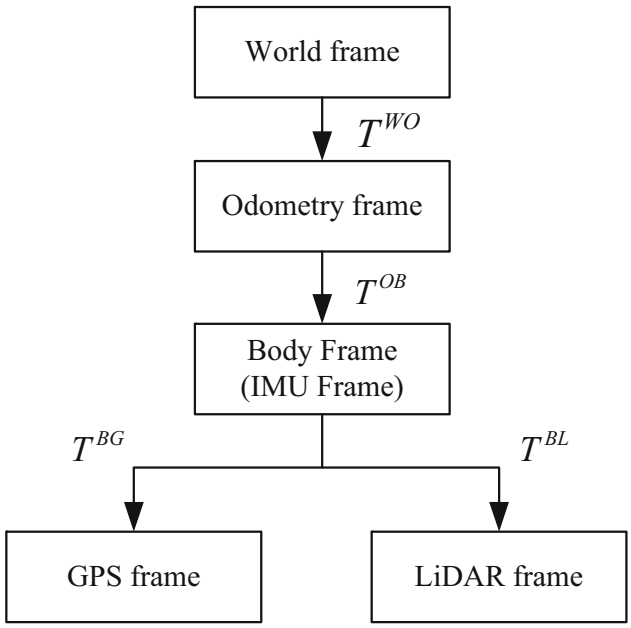


Fig. 2 The Frame Tree of the Proposed System

The transformation from frame **B** to frame **A** can be denoted as $(\cdot)^{AB}$, and it is worth mentioning that the T from the reference frame to the object frame can also describe the object's pose in the reference frame, so we have $T = [R|p]$. The frame tree of our system can be visualized in Fig. 2, where T^{BG} and T^{BL} denote the extrinsic parameters of GPS and LiDAR to IMU, and we calibrate them in advance.

Because most of our work analyzes the platform in the odometry frame **O**, we omit the superscript OB for convenience if it is evident in context. For example, we use T to denote T^{OB}.

3.2 System Structure

This paper presents a robust LiDAR SLAM algorithm with a ZUPT-aided initialization procedure for 6-DOF state estimation. The system pipeline is shown in Fig. 3.

The system consists of three parts: initialization procedure, LIO, and global optimization, where our main contribution is the initialization procedure (blue area in Fig. 3). The initialization procedure starts at system startup and ends after the platform has driven for a specific time. Under the assumption that the platform is always bootstrapped from stationary, we divide the initialization procedure into two phases: static and dynamic. We also propose a Zero Velocity Detection method to distinguish the two phases. The primary purpose of the initialization procedure is to estimate the unknown initial state, including the initial pose and the IMU bias, to ensure the system can robustly transit to normal operation (LIO and global optimization). The LIO combines the LiDAR odometry with the IMU preintegration for state

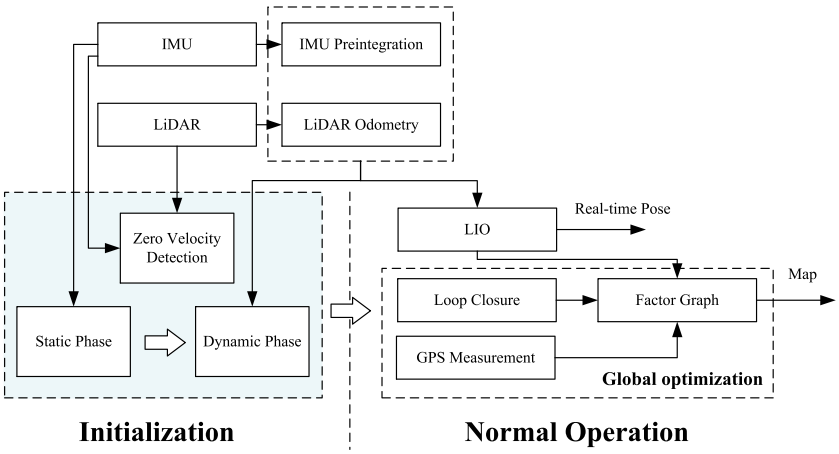


Fig. 3 The System Pipeline. The system includes an initialization procedure and a normal operating procedure. The main contribution of our work is the initialization procedure, which we mark in the figure blue. Under the assumption that the platform is always bootstrapped from stationary, we divide the initialization procedure into two phases: static and dynamic. The system runs in the order of static initialization, dynamic initialization, and normal operation. We also propose a Zero Velocity Detection method to distinguish the two initialization phases. The LIO and the global optimization run in the normal operating procedure. The LIO provides the state estimation. The global optimization built atop the pose graph provides the transformation from odometry frame **O** to world frame **W**, and it also corrects accumulated error

estimation. The global optimization is built atop the pose graph, and obtains the LiDAR odometry factor, the GPS factor, and the loop closure factor for optimization, which results in a global consistency map.

4 Measurement Preprocessing

Due to space limitations, this section briefly introduces the LiDAR odometry and IMU preintegration method followed by our work. Please refer to the origin paper for detail.

4.1 LiDAR Odometry

Our LiDAR odometry follows the work of LOAM in [4] and LIO-SAM in [3]. The LOAM proposed a feature-based scan-matching method. When the LiDAR frame $\mathbb{F}_n$ is received, it first extracts the edge and planar feature from the LiDAR scan by evaluating the roughness of points over a local region. They are defined as $\{F_n^e, F_n^p\}$ in LiDAR frame **L**. Then, the extracted features are matched to the features from previous frames. The distance of matched features is as follow:

$$\left\{d_e(\tilde{T}_n^{OL} p_{n,i}^e), d_p(\tilde{T}_n^{OL} p_{n,j}^p)\right\} \tag{3}$$

where $p_{n,i}^e \in F_n^e$, $p_{n,j}^p \in F_n^p$. The $\tilde{T}_n^{OL}$ is the pose of **L** in **O**, and it is to be estimated.

The optimization problem is to minimize the sum of the distance from matched features. Through manifold optimization [26] and GaussNewton method, we can solve it to obtain the transformation $\hat{T}_n^{OL}$. Given the extrinsic parameters between **L** and **B** as T^{BL}, then we can calculate the pose of frame **B** as $\hat{T}$ and the relative transformation $\Delta\hat{T}_{n-1,n}$ between the $\mathbb{F}_{n-1}$ and $\mathbb{F}_n$. So the residual errors of two frames can be defined as:

$$r_L = \left[\delta\xi_{n-1,n}\right] = log((\Delta T_{n-1,n})^\top \Delta\hat{T}_{n-1,n})^\vee \tag{4}$$

Based on the LOAM, the LIO-SAM proposed a scan-to-sub-map method instead of the scan-to-scan method, improving the accuracy. In addition, the latter work adopts the keyframe selection method for reducing the computation. It registers a LiDAR frame as a keyframe when its pose exceeds a user-defined threshold compared with the previous keyframe. In the following parts of this paper, we will also focus on keyframes. Therefore we redefine the nth LiDAR keyframe as $\mathbb{F}_n$ and the related state as X_n.

4.2 IMU Preintegration

We follow the IMU preintegration method proposed in [13] to obtain the relative platform motion between two LiDAR frames. The standard IMU integration method needs repeat calculation due to the states changing at each optimization iteration. The IMU preintegration avoids this by decoupling the estimating states from the IMU measurements, significantly improving the real-time performance.

The residual errors between two LiDAR frames (X_{n-1}, X_n) can be written as:

$$r_I = \begin{bmatrix} \delta\phi_{n-1,n} \\ \delta v_{n-1,n} \\ \delta p_{n-1,n} \\ \delta b^a_{n-1,n} \\ \delta b^\omega_{n-1,n} \end{bmatrix} = \begin{bmatrix} log((\Delta R_{n-1,n})^\top \Delta \hat{R}_{n-1,n})^\vee \\ \Delta \hat{v}_{n-1,n} - \Delta v_{n-1,n} \\ \Delta \hat{p}_{n-1,n} - \Delta p_{n-1,n} \\ b^a_n - b^a_{n-1} \\ b^\omega_n - b^\omega_{n-1} \end{bmatrix}. \quad (5)$$

where $\Delta R_{n-1,n}, \Delta v_{n-1,n}, \Delta p_{n-1,n}$ denote the preintegration items, $\Delta \hat{R}_{n-1,n}, \Delta \hat{v}_{n-1,n}, \Delta \hat{p}_{n-1,n}$ denote the measurements of preintegration items, and $\delta\phi_{n-1,n}, \delta v_{n-1,n}, \delta p_{n-1,n}$ denote the noise of them respectively. $\delta b^a_{n-1,n}$ and $\delta b^\omega_{n-1,n}$ denote the bias instability. The detailed definition can be seen in the original paper.

5 Tightly-coupled LiDAR Inertial Odometry

In order to draw out the significance of this paper, this section introduces the LIO method and illustrates the problems of the LIO method without the initialization process.

We build the LIO procedure atop a window-based factor graph optimization. Assumed that the window capacity is N, the entire state vector in the sliding window is:

$$\chi_n = [X_{n-N+1}, X_{n-N+2}, \cdots, X_n]. \quad (6)$$

The optimization graph and the pipeline are shown in Fig. 4. In general, the sampling frequency of IMU is higher than LiDAR. Therefore, based on the previous states X_{n-1}, we can obtain the IMU preintegration value to estimate the T_n before the LiDAR scan arrives. Then, when the LiDAR scan arrives, the IMU preintegration value can be utilized for LiDAR raw data de-skewing and regarded as the initial guess for LiDAR scan matching optimization. Through the scan match, the LiDAR odometry can be obtained. Finally, the LiDAR odometry and the IMU preintegration are jointly optimized atop a sliding window-based factor graph. We minimize the sum of measurement residuals of IMU and LiDAR odometry to obtain a maximum posteriori estimation as:

$$\min_{\chi} \{\|r_M(\chi)\| + \sum_{k\in\tau} \left\|r_I(z^I_k, \chi)\right\|^2 + \sum_{k\in\beta} \left\|r_L(z^L_k, \chi)\right\|^2\} \quad (7)$$

where r_M is the prior items form marginalization. r_I and r_L are residuals from IMU preintegration and LiDAR odometry, which are defined in (5) and (4). τ and β denote the IMU measurements and transformation, respectively. Then the optimized state X_n will be used at the next iteration.

Although the LIO can perform well, it also suffers from unknown initial states when bootstrapping the system, including IMU bias b_0 and posture R_0.

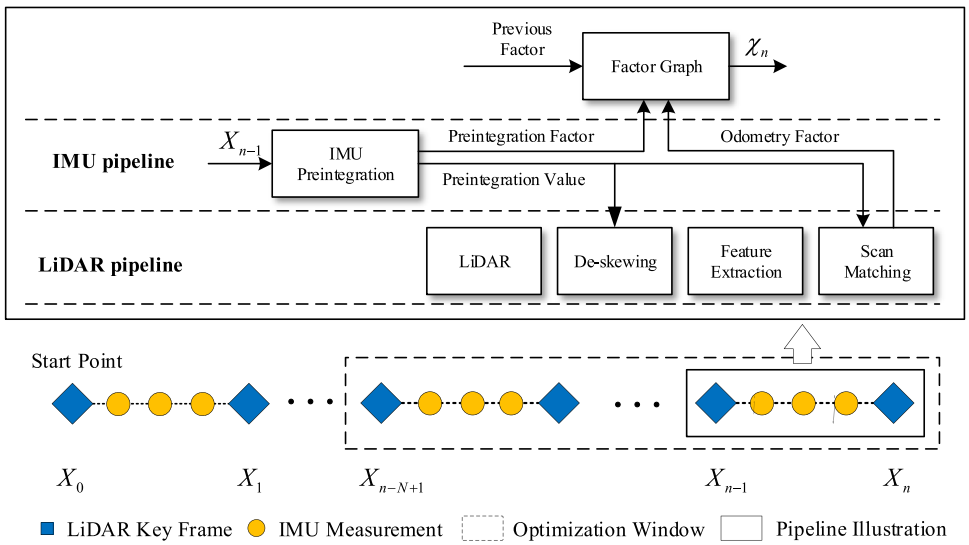


Fig. 4 LIO Pipeline. At each iteration, the IMU preintegration value, based on the previous states X_{n-1}, will be utilized for LiDAR raw data de-skewing and regarded as the initial guess for LiDAR scan matching optimization. Then, the LiDAR odometry and the IMU preintegration are jointly optimized atop a sliding window-based factor graph. However, when the platform is bootstrapped from stationary with the unknown initial state, the IMU preintegration value is imprecise, resulting in imprecise LiDAR odometry. In addition, the imprecise LiDAR odometry and the unknown initial states will deteriorate the conjunction optimization in (7), providing poor results of current states, which will affect the next LIO iteration. As the LIO pipeline progress, the accumulated error will cause the algorithm divergence

Firstly, the point cloud features cannot be matched exactly due to the sparsity, especially in the vertical direction. As a result, the initial guess of scan matching will extraordinarily affect the accuracy of the result. However, the preintegration value will be imprecise under the rough artificial estimation of the unknown initial state X_0. Therefore, utilizing it for LiDAR raw data de-skewing and acting as the initial guess will quickly trap the scan matching optimization into a local minimum, which will cause imprecise LiDAR odometry.

Secondly, the imprecise LiDAR odometry and the unknown initial states will deteriorate the conjunction optimization in (7), providing poor results of current states.

Finally, as shown in Fig. 4, the current state will affect the next LIO iteration, and the accumulated error will cause the algorithm divergence with the pipeline progress. This error will be amplified with low-cost sensors or application scenarios with fewer features, such as indoor or open-scene environments.

We carried out a simple experiment to demonstrate the divergence of the LIO method with unknown initial states in indoor and outdoor scenarios. The experiment adopted the low-cost 6-axis IMU and 16-rings LiDAR. We placed them together in the experimental scene and kept them static. Then, we let the system bootstrap with unknown initial states. The operation result is shown in Fig. 5. Figure 5(a) and (d) illustrates the indoor and outdoor environments. We chose a relatively empty area to demonstrate our point. Figure 5(b) and (e) illustrate the operation result of the LIO method displayed in RVIZ. The ghost image of LiDAR points demonstrates that the LIO method is divergent both in the indoor and outdoor environment when the platform is stationary. Figure 5(c) and (f) illustrate the translation error in the horizontal plane. We can see that translation error rapidly accumulates, especially in the outdoor environment where the LiDAR feature point is lower than the indoor, and the curves also confirm the divergence.

6 Initialization Procedure

6.1 Structure of Initialization Procedure

Motivated by the initialization procedure in VINS, the famous visual SLAM proposed in [17], we design an initialization procedure to obtain the unknown initial states for LIO. Figure 6 shows the pipeline block diagram and the graph of the initialization procedure.

Assuming that the platform is always bootstrapped from stationary, our initialization procedure starts at the beginning of the system and ends after a fixed number of N times LiDAR keyframes are registered. So the states in the initialization procedure can be written as $[X_0, X_1, \cdots, X_N]$. Where $X_0 = [{R_0}^\top, {p_0}^\top, {v_0}^\top, {b_0}^\top]^\top$ denotes the state when the platform is stationary. Since the initial position defines the position origin of the odometry frame O, we have $v_0 = [0, 0, 0]^\top$ and $p_0 = [0, 0, 0]^\top$. The keyframes number N in the initialization procedure can be adjusted according to the actual situation. In our work, it is the same as the window size of LIO (see Section 5).

Our initialization procedure includes two phases: static phase and dynamic phase. We also design the Zero Velocity Detector to distinguish the phases. When the system is bootstrapped, the Zero Velocity Detector and static phase begin to work. The Zero Velocity Detector utilizes two measurements to detect the stationary. The one is a fixed set time W of consecutive IMU measurements, and we denote it as ψ. The other is the residual error of the LiDAR scan match. The static phase also utilizes IMU measurements in ψ to estimate the R_0 and b_0 by Zero Velocity Update. The estimation result in the static phase will be used as the prior factor in the dynamic phase. When the Zero Velocity Detector determines the platform is in a move, the Zero Velocity Detector and the static phase end. Meanwhile, the dynamic phase starts to work. The main idea of the dynamic phase is to utilize the LiDAR odometry as a baseline and align the IMU preintegration with it for optimization. So we do not optimize the rotation R and position p of states except X_0. As a result, the unknown initial states that need to be obtained can be defined as:

$$\chi_{init} = [R_0, b_0, v_1, b_1, v_2, b_2, \cdots, v_N, b_N]. \tag{8}$$

6.2 Zero Velocity Detection

The proposed Zero Velocity Detection method is built by combining the inertial-based and LiDAR-based detection methods. The inertial-based detection typically utilizes the generalized likelihood ratio test (GLRT) [27] for IMU measurements to detect the stationary. The algorithm steps are as follows:

Taking the accelerator bias b^a, gyroscope bias b^ω and the additive noise into consideration, the IMU measurements formula can be written as follow:

$$\begin{aligned} \hat{a} &= R^\top (a - g) + b^a + n^a \\ \hat{\omega} &= \omega + b^\omega + n^\omega. \end{aligned} \tag{9}$$

where t denote one sampling time. $\hat{a} \in \mathbb{R}^3$ and $\hat{\omega} \in \mathbb{R}^3$ denote the measurements of accelerator and gyroscope in frame **B** respectively. ω is the instantaneous velocity of platform expressed in frame **B**, while a is the acceleration in frame **O**. $g = g^W = g^O = [0, 0, g]^\top$ denote the gravity vector in frame **O**. Due to the IMU frame being defined as a body frame, the R also denotes the rotation of IMU.

The n^a and n^ω denotes the additive noise in acceleration and gyroscope measurements, which can be seen as the Gaussian white noise as $n^a \sim N(0, \sigma_a^2)$, $n^\omega \sim N(0, \sigma_\omega^2)$.

The GLRT value $\mathbb{G}$ of the IMU data can be written as:

$$\mathbb{G} = \frac{1}{W}\sum_{k\in\psi}\left(\frac{1}{\sigma_a^2}\left\|\hat{a}_k - g\frac{\bar{a}_k}{\|\bar{a}_k\|}\right\|^2 + \frac{1}{\sigma_\omega^2}\|\hat{\omega}_k\|^2\right). \tag{10}$$

where the $\bar{a}$ denotes the sample mean:

$$\bar{a} = \frac{1}{W}\sum_{k\in\psi}\hat{a}_k \tag{11}$$

According to hypothesis testing theory, the higher the GLRT value, the more likely the platform will move.

The LiDAR-based detection method utilizes the mean residual error of LiDAR scan matching to detect the stationary. When the system is bootstrapped and maintained stationary, the sub-map M (Defined in Section 5) only contains the features of the first LiDAR scan. So we utilize the current LiDAR scan to match the first LiDAR scan for calculating the distance of features through Eq. (3). Then, the mean residual error of LiDAR scan matching can be defined as:

$$\mathbb{E} = \frac{1}{N_e}\sum_{p_{n,i}^e \in F_n^e} d_e(p_{n,i}^e) + \frac{1}{N_p}\sum_{p_{n,j}^p \in F_n^p} d_p(p_{n,j}^p). \tag{12}$$

By combining the $\mathbb{G}$ and the $\mathbb{E}$, we can detecte the stationary as:

$$\text{ZUD} = \begin{cases} Stationary & (\mathbb{G} \le \theta \; or \; \mathbb{E} \le \gamma) \\ Moving & (\mathbb{G} > \theta \; and \; \mathbb{E} > \gamma) \end{cases} \tag{13}$$

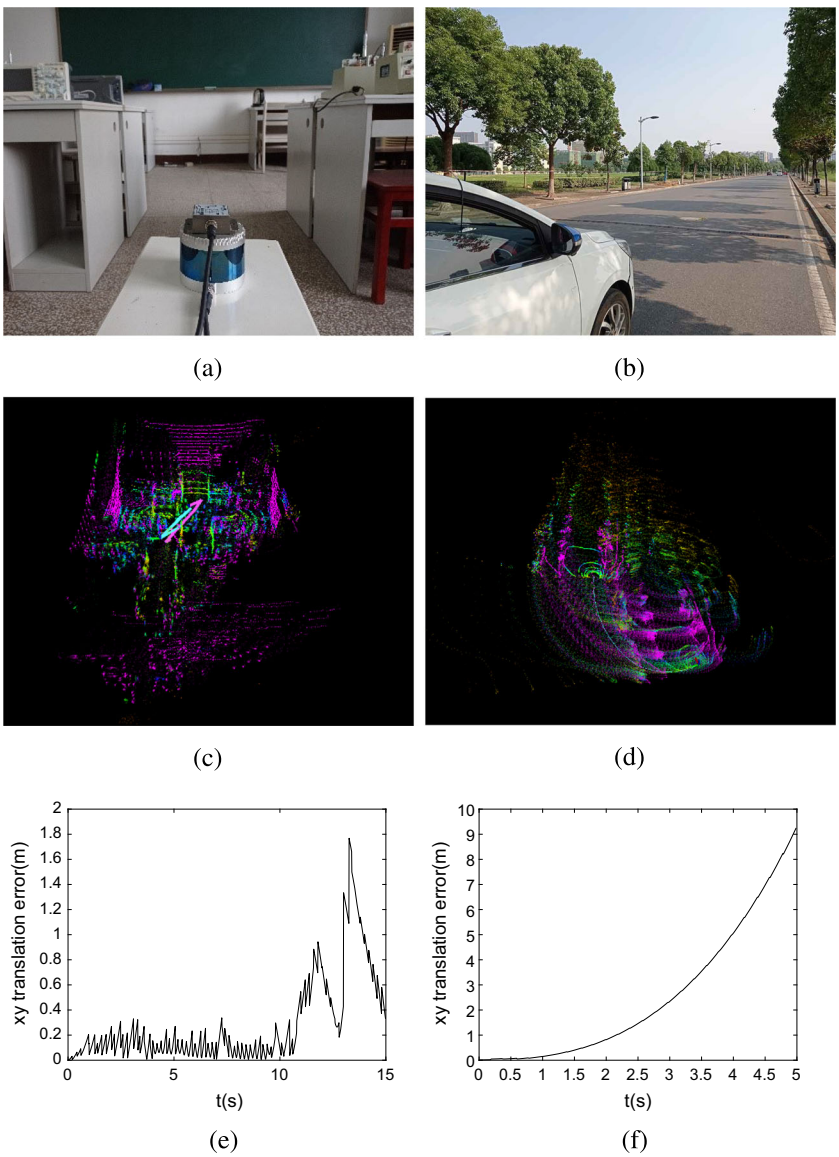


Fig. 5 Static Experiment: **a** Indoor environment. **b** Outdoor environment. **c** Operation result in RVIZ for indoor experiment. **d** Operation result in RVIZ for outdoor experiment. **e** Translation error for indoor experiment. **f** Translation error for outdoor experiment

where the θ and the γ are the threshold value of the $\mathbb{G}$ and the $\mathbb{E}$, respectively. The threshold values are determined through experimental means. Among these values, the parameter $\mathbb{G}$ has no units and there is a significant difference between static and dynamic states. When in a static state, the ideal value of $\mathbb{G}$ is 2 (however, due to the influence of IMU bias and environmental noise, this value tends to be larger in practice), and thus θ is set to a range of 10 to 100. The parameter γ is typically set between 120% $\sim$ 150% of the static value of $\mathbb{E}$. The detection results are used to distinguish between the two initialization phases.

6.3 Static Phase

In the static phase, We utilize the last measurements ψ before the end of the static phase to estimate the R_0 and b_0 by Zero Velocity Update.

Due to the stationary constraint during that time, the true acceleration and angular velocity are zero, and the posture of the platform remains unchanged. We also assumed that the bias remains constant. So we have:

$$\begin{aligned} a_k &= 0 \\ \omega_k &= 0 \\ R_k &= R_0 \\ b_k &= b_0. \end{aligned} \tag{14}$$

Frome the Eq. 9, we have the measurement function as:

$$\begin{aligned} \hat{a_k} &= -{R_0}^\top g + b_0^a + n_k^a \\ \hat{\omega_k} &= b_0^\omega + n_k^\omega. \end{aligned} \tag{15}$$

Thus, we can easily obtain the following residual error:

$$\begin{aligned} r_{zvu} &= \begin{bmatrix} n_k^a \\ n_k^\omega \end{bmatrix} = \begin{bmatrix} \hat{a_k} + {R_0}^\top g - b_0^a \\ \hat{\omega_k} - b_0^\omega \end{bmatrix} \\ &= \begin{bmatrix} \hat{a_k} + \exp\left((-\phi_0)^\wedge\right) g - b_0^a \\ \hat{\omega_k} - b_0^\omega \end{bmatrix} \end{aligned} \tag{16}$$

The optimization state is denoted as $\chi_{sp} = \{\phi_0, b_0^a, b_0^\omega\}$, and the optimization goal is:

$$\min_{\chi_{sp}} \sum_{\psi} \|r_{zvu}\|^2 \tag{17}$$

Moreover, we have the following Jacobians matrix:

$$J = \begin{bmatrix} (R_0^\top g)^\wedge & -\mathrm{I}_{3\times 3} & 0 \\ 0 & 0 & -\mathrm{I}_{3\times 3} \end{bmatrix} \tag{18}$$

Through the Gauss-Newton iteration method and optimization packages such as ceres [28] or iSAM [29], we can solve this problem and obtain the estimation of χ_{sp}. Then, we will utilize the state as the prior value in the dynamic phase.

6.4 Dynamic Phase

Once the vehicle is ready to move, the Zero Velocity Detection and static phase end. Meanwhile, the dynamic phase begins to work. In this process, we align the IMU preintegration to the LiDAR odometry for estimating the χ_{init}. It is worth noticing that there are two differences in LiDAR odometry between this process and the LIO process (See Sec. 5):

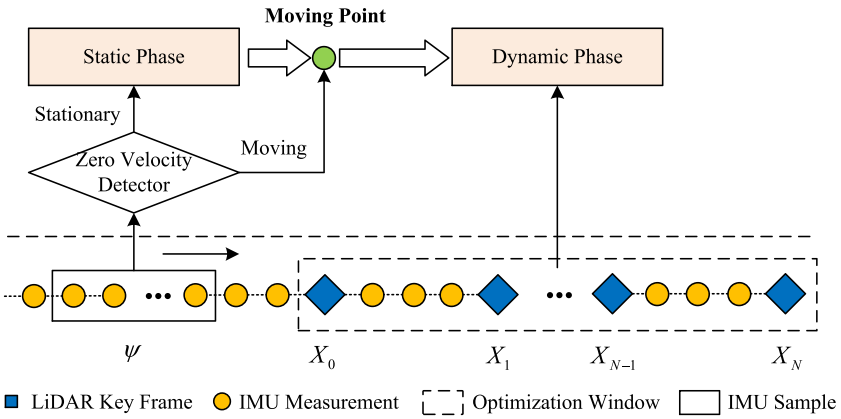


Fig. 6 Initialization Pipeline. Assuming that the platform is always bootstrapped from stationary, our initialization procedure starts at the beginning of the system. It ends after a fixed number of N times LiDAR keyframes are registered. The overall procedure includes the static phase and dynamic phase. We also design the Zero Velocity Detector to distinguish the phases. When the system is bootstrapped, the Zero Velocity Detector and static phase begin to work. When the Zero Velocity Detector determines the platform is in a move, the Zero Velocity Detector and the static phase end, meanwhile, the dynamic phase starts to work

- First, because the IMU preintegration is imprecise with the unknown initial states, we utilize the inertial estimation to obtain the initial guess for scan matching in this process. The platform's velocity is low during that time, so there is only a slight error for the initial guess.
- Second, we reduce the threshold of registering the keyframes because there are few keyframes in the sub-map window. As a result, the features in the sub-map can rapidly increase, which will benefit the precision of LiDAR odometry.

Assumed that the LiDAR poses in **O** is T_n^{OL} and given the extrinsic parameters between **L** and **B** as T^{BL}, the transformation of the platform can be written based on a progressive relationship as:

$$\begin{aligned} T_n &= T_n^{OL}(T^{BL})^{\top} \\ &= T_0^{OL}\Delta T_{0,1}^{L}\Delta T_{1,2}^{L}\cdots\Delta T_{n-1,n}^{L}(T^{BL})^{\top} \\ &= T_0(T^{BL})\Delta T_{0,1}^{L}\Delta T_{1,2}^{L}\cdots\Delta T_{n-1,n}^{L}(T^{BL})^{\top}. \end{aligned} \tag{19}$$

where the $p_0 = [0, 0, 0]^{\top}$ and the transition between two LiDAR scan $\Delta T_{n-1,n}^{OL}$ can be obtained from LiDAR odometry in Sec. 4.1, so the T_n is only involves with R_0. Then we substitute the R_n, p_n for IMU preintegration items in (5) and get the residual error function as follwos:

$$\min_{\chi_{init}} \{\left\| r_\rho(\chi_{init}) \right\| + \sum_{k\in\beta} \left\| r_I(z_k^L, \chi_{init}) \right\|^2\} \tag{20}$$

where the $r_\rho(\chi_{init})$ denote the priori value of $[R_0, b_0]$ from the estimation in (16). r_I denote the residuals of IMU preintegration in (5). Through the optimization by the Ceres solver [28], the states χ_{init} can be obtained, and it can be used in proceeding after initialization. The overall initialization procedure can be represented in pseudocode as Algorithm 1.

Algorithm 1 Initialization Procedure

```
SystemState = StaticInit
while n < N do
  if SystemState == StaticInit then
    if A now IMU data received then
      Calculate 𝔾 from Eq. (10)
    end if
    if A now LiDAR data received then
      Calculate 𝔼 from Eq. (12)
      Zero velocity detected through Eq. (13).
      if Stationary then
        Zero velocity update.
        Set the prior value of [R_0, b_0]
      elseIf Moving
        SystemState = DynamicInit
      end if
    end if
  elseIf SystemState == DynamicInit
    if A now IMU data received then
      IMU preintegration.
    end if
    if A now LiDAR keyframe registered then
      n = n + 1
      LiDAR odometry.
    end if
    if n = N then
      Dynamic phase optimization from Eq. (20)
    end if
  end if
end while
```

7 Global Optimization

Global optimization is irrelevant to our main contribution, so we only provide a brief introduction. LIO can get accurate odometry. However, there are still two issues that remain. First, the system still suffers from drift during long-duration navigation tasks. Second, the transformation between the world and odometry frames T^{WO} still needs to be estimated. As a result, we design a global optimization to handle those problems.

The global optimization utilizes the GPS factor and loop closure factor to estimate the T^{WO} and eliminate the drift of LIO, and it is built atop the pose graph. We utilize the Scan Context descriptor for detecting the loop closure [30], and we project the raw GPS data from the WGS-84 coordinate to ENU coordinate to unify the LiDAR odometry and GPS scale. The detail of the pose graph method can be found in paper [5].

8 Experiments

8.1 Datasets Preparation

To demonstrate the efficiency of our method, we collected several sets of data on the WHUT campus and carried out experiments. We utilize an electric vehicle as the platform in our experiment and equip it with a low-cost sensor suite, including an ouster OS1-16 and a vigor technology TT810 IMU module. The platform is equipped with an RTK GPS module to provide ground truth. We install the LiDAR ahead, the IMU and the RTK GPS inside, and the antennas of the RTK GPS at the top. The platform is shown in Fig. 7. We calibrate the internal parameters of IMU by the method proposed in [31], and the value is shown in Table 2.

Our method is implemented in C++ and executed on a JETSON TX2 with the framework of Robot Operating System (ROS) [32] in Ubuntu Linux.

Fig. 7 The Experimental Platform Illustration. The Electric vehicle is used as the platform. The LiDAR OS1-16 is installed ahead of the vehicle. The IMU TT810 and RTK GPS are installed and fixed inside the vehicle. The antennas of RTK GPS are installed atop the vehicle. The start point environment of each dataset

For the convenience of verification, we separately used different collected datasets to verify Zero Velocity Detection and initialization procedure.

8.2 Zero Velocity Detection Evaluation

We evaluated the performance of the proposed Zero Velocity Detector by collecting three datasets under different conditions. ZVD A was tested under normal start-up conditions, ZVD B was subjected to vibration interference, and ZVD C was subjected to dynamic object interference. I compared our method with the GLRT method and the MRE method. We recorded the GLRT $\mathbb{G}$ and the mean residual error $\mathbb{E}$ (see Section 6.2) marked MRE. We also recorded the horizontal moving distance estimated by GPS as a reference, and we took the moment that the platform moved 0.1 meters as the reference moving point. Their sample frequency is the same as LiDAR frequency, and we utilized the IMU data between two LiDAR frames to calculate the GLRT. The curves of GLRT, MRE, and GPS Trans are shown in the Fig. 8. Figure 8(a)-(c) show the data graph of collection datasets ZVD A, B, and C, respectively. The analysis of the figures is as follows.

The trend of the curves in Fig. 8(a) shows that both GLRT and MRE can effectively reflect the stationary state of the system. However, as shown in Fig. 8(b), GLRT is easily disturbed by vibrations, leading to significant perturbations. Therefore, the simple GLRT method based on IMU data is susceptible to interference and may cause misjudgment in complex environments. In contrast, MRE is less affected by vibration interference. Nevertheless, as shown in Fig. 8(c), the MRE curve has a significant bulge when the platform is stationary due to interference from dynamic objects, which can affect the accuracy of the LiDAR-based detection method. Moreover, as the number of dynamic objects in the environment increases, the dynamic interference on MRE's judgment of the stationary state will also gradually increase. Therefore, our proposed method, which combines LiDAR-based and inertial-based detection methods, can significantly reduce the possibility of misestimation.

Table 2 IMU internal parameter

parameter	units	value
Gyroscope "white noise"	$\frac{rad}{s}$	2.15e-03
Accelerometer "white noise"	$\frac{m}{s^2}$	3.74e-02
Gyroscope "bias Instability"	$\frac{rad}{s}$	8.03e-05
Accelerometer "bias Instability"	$\frac{m}{s^2}$	2.84e-03

To further quantitatively compare the performance, we set the threshold of GLRT to 20 and the threshold of MRE to 0.06 for their estimations of the moving points. The estimated times of the moving points and the reference moving points are listed in Table 3. It can be observed that, due to GLRT's higher sensitivity compared to MRE, our estimate of the moving point remains consistent with MRE when there is only a small amount of dynamic object interference in the environment. Our method enhances robustness without compromising the accuracy in determining the moving point. The proposed Zero Velocity Detector effectively detects the moving point.

8.3 Startup Evaluation

We collected three datasets in a different environment of the WHUT campus to evaluate the initialization procedure. All of them are collected when there are fewer dynamic objects in the environment to avoid other interference. In datasets A and B, the vehicle starts from static and drives straight for a while. In Datasets C, the vehicle starts from static, then turns right immediately. The details of these datasets are shown in Table 4. The start point environment of each dataset is shown in Fig. 9(a)-(c) respectively. We can see that the start point of datasets A and C are located in an environment with abundant features, and there is a building on the left. In contrast, dataset B's start point has fewer features, and there are only a few trees in the environment.

Firstly, we utilize the IMU raw data as input to estimate the states $\left[R_0, b_0^{\omega}\right]$ for evaluating the static phase of the initialization procedure. The IMU measurements are between two LiDAR frames, and the number is 10. We recorded the last estimation before the end of the static phase and converted

Table 3 The estimation result of moving point (s)

Dataset	Reference	GLRT		MRE		our	
		Estimated	Time Error	Estimated	Time Error	Estimated	Time Error
ZVD A	38.47	38.07	-0.40	38.77	0.29	38.77	0.29
ZVD B	33.98	17.99	-15.99	33.38	-0.59	33.38	-0.59
ZVD C	34.57	34.25	-0.32	34.66	0.09	34.66	0.09

Table 4 Dataset details

Dataset	Trajectory length (m)	Max Speed (Km/h)	Time period
Collection A	105.98	23.10	30
Collection B	137.88	28.5	30
Collection C	179.59	23.7	40
Campus	2706.34	28.78	494

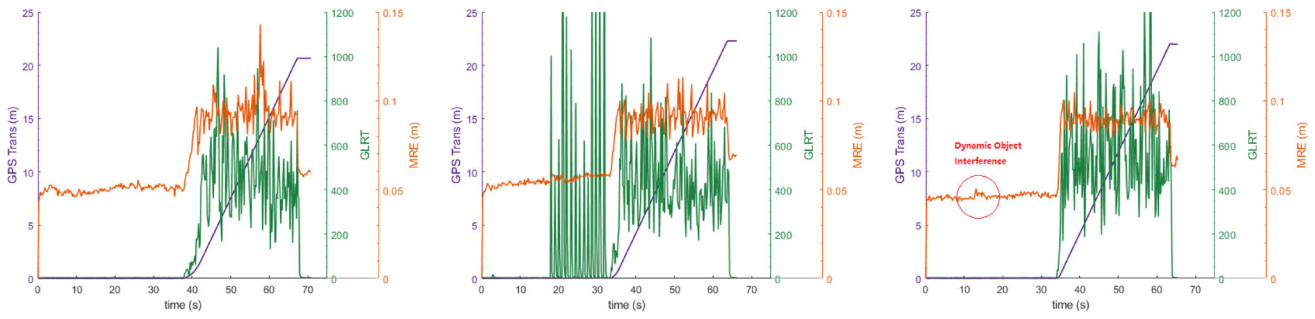
Table 5 The estimation results of unknown initial states

Dataset	roll (rad)	pitch (rad)	$b^a(m/s)$	$b^\omega(rad/s)$
Collection A	-0.0391	0.0364	[0.0042; 0.0045; −0.0738]	[−0.0246; −0.0032; 0.0128]
Collection B	-0.0385	0.0318	[0.0036; 0.0043; −0.1010]	[−0.0251; −0.0026; 0.0126]
Collection C	-0.0347	0.0342	[0.0039; 0.0039; −0.1369]	[−0.0251; −0.0028; 0.0123]

the estimated transformation matrix to Euler angles to make it more intuitive. The result is shown in Table 5.

Then, we carried out a startup experiment to evaluate the benefit of our initialization procedure for localization performance at startup. We compare the proposed method with four references. One of them is **LIO-SAM**. The second is LIO-SAM with a simple initialization method based on calculating the states using the mean of the IMU measurements, and we mark it **LIO-SAM-mean**. The third one is only the static phase initialization procedure, and we mark it **our-static**. The last one is only the dynamic phase procedure, and we mark it **our-dynamic**. We also mark the complete proposed method as **our**.

We used the RTK signal as the ground truth, and record the Max Trans Error and RMSE Trans Error in the horizontal direction to evaluate the precision. We also record the number



(a) (b) (c)

Fig. 8 The Result of Zero Velocity Detection Experiment. We record the GLRT $\mathbb{G}$, the mean residual error $\mathbb{E}$, and the moving distance estimated by GPS. Where the MRE denotes the $\mathbb{E}$. In order to intuitively show the trend of the three quantities at startup, we put the three curves in the same figure. The blue line indicates the GPS distance, the red line indicates the $\mathbb{E}$, and the green line indicates the $\mathbb{G}$. **a** The result of dataset A. **b** The result of dataset B. **c** The result of dataset C

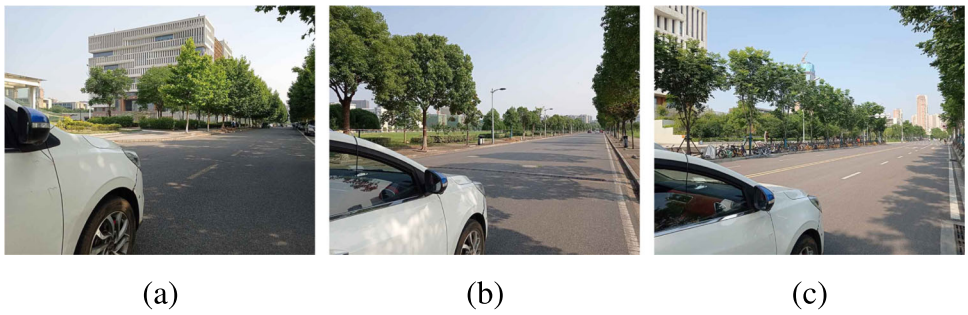


Fig. 9 The Experimental Environment Illustration. **a** The environment of data set A. **b** The environment of data set B. **c** The environment of data set C

Table 6 Startup experiment result

Dataset	Features	LIO-SAM		LIO-SAM-mean		our-static		our-dynamic		our	
		Max	RMSE	Max	RMSE	Max	RMSE	Max	RMSE	Max	RMSE
Collection A	1788	Fail		Fail		Fail		0.416	0.222	**0.409**	**0.218**
Collection B	1238	Fail		Fail		Fail		0.440	0.230	**0.379**	**0.209**
Collection C	2453	0.602	0.290	**0.562**	0.269	0.570	0.265	0.577	0.261	0.569	**0.256**

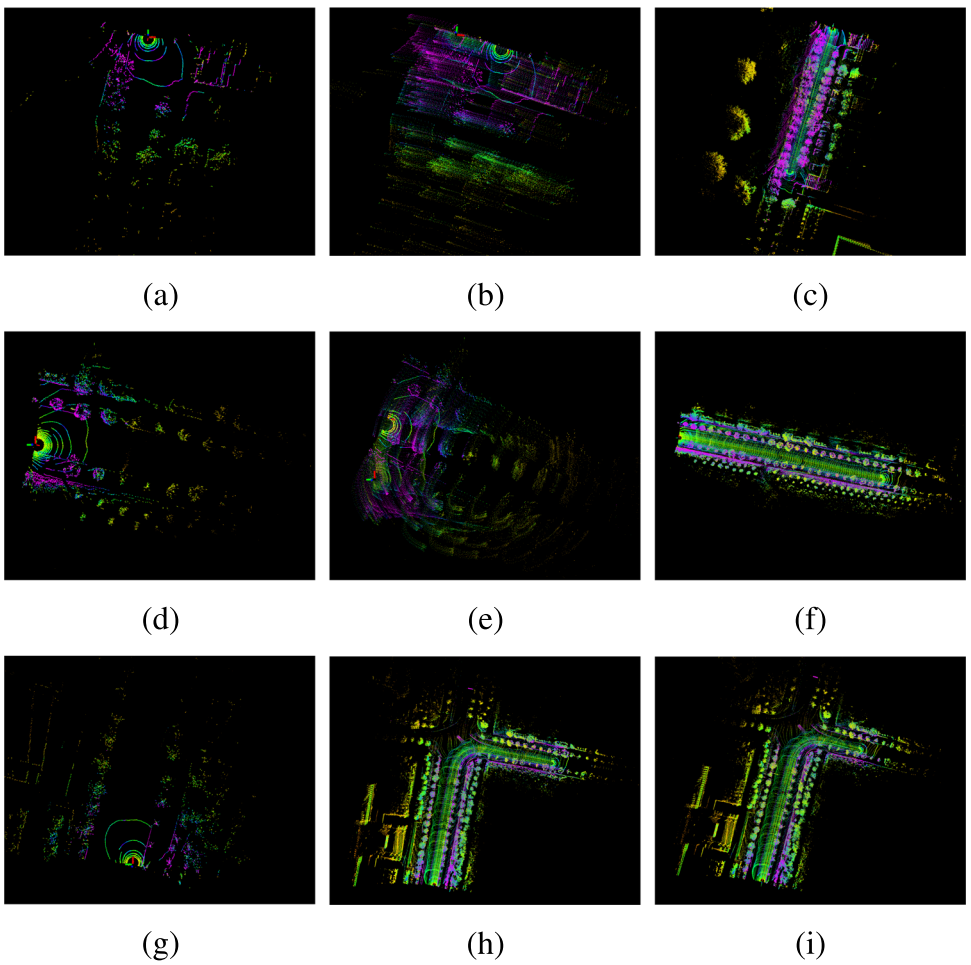


Fig. 10 Startup Experimental Results Illustration. **a**, **d**, **g** shows the first scan of datasets A, B, and C, respectively. (b) (e) (h) shows the startup result of LIO-SAM. (c) (f) (i) shows the startup result of our. As is shown, the LIO-SAM fails to startup due to the interference of the unknown initial state in datasets A and B. In contrast, our method with an initialization procedure can overcome this problem and provide an expected result in all datasets

of feature points, indicating the richness of environmental features. The result is shown in Table 6. The data highlighted in bold report the best results.

For more intuition, we illustrate the operation result figures of LIO-SAM and our method in Fig. 10. We also illustrate the first LiDAR scan figures, which show the sparsity of features. These figures are all from RVIZ, a visualization tool of ROS.

Firstly, we compare our method with LIO-SAM. We can see that in the A and B datasets, the LIO-SAM fails to startup due to the interference of unknown initial state. The ghost image of LiDAR points in the Fig. 10(b) and (e) also demonstrate that the LIO-SAM deviated at the beginning when the vehicle is static due to few feature points. The LIO-SAM can normally startup in dataset C. From the table, we can also see that dataset C has more feature points than others, so it has enough feature points to restrain the drift, and its result is shown in Fig. 10(e). In contrast, our method with an initialization procedure can overcome this problem and provide a normal result in all datasets. It indeed improve the robustness of the system, the Fig. 10(c), (f), (i) shown the operation result figure of our method. However, when the LIO-SAM can normally start, like dataset C, we can see that our method partially improves precision because the LiDAR odometry contributes more precision when the feature points are enough to restrain the drift.

Then, the result of LIO-SAM-mean shows that the simple initialization procedure cannot overcome this problem. Although compared with LIO-SAM in dataset C, it has a specific improvement in precision, thanks to its estimation of initial posture.

Finally, we compare our method with the our-static and our-dynamic. The our-static also fails to startup in the A and B datasets, so the dynamic initialization phase is more contribute to the system. However, comparing the precision of our-dynamic and our, we can see that the static phase can partly improve the precision because the dynamic initialization phase cannot precisely estimate the initial posture.

8.4 Runtime Evaluation

In this section, We utilize the datasets to test the time consumption of Zero Velocity Detection, static phase, and dynamic phase, respectively. It should be noted that although Zero Velocity Detection utilizes the mean residual error of LiDAR scan matching, the calculation of residual error belongs to scan-matching, so the time consumption of this part is not added to the Zero Velocity Detection. The result is shown in Table 7. As is shown, the time consumption of overall initialization procedure is not large, and the scan-matching more significantly influences the runtime of the system.

Table 7 Runtime Result (ms)

Dataset	Zero Velocity Detection	Static Phase	Dynamic Phase
Collection A	0.048	0.006	3.51
Collection B	0.035	0.007	3.34
Collection C	0.057	0.007	2.25

9 Conclusions and Discussion

Inthis paper, we proposed a Zero Velocity Update (ZUPT) aided initialization procedure to estimate unknown initial states, including the initial pose and the IMU bias, to ensure the tightly coupled LiDAR and IMU odometry system can robustly startup. The proposed method has a better effect on low-cost sensor suit and significantly improve the robustness of scenes with sparse features or significant IMU bias and noise. Because of the initial posture estimation through the static phase, our method can be applied to the 6-axis IMU. The proposed system is evaluated on the datasets gathered from the campus zone. The results show that our approach can effectively improve robustness and partly improve precision when the system is bootstrapped.

In the subsequent work, we will continue combining some filtering or diagnostic algorithms for IMU with the LIO-based SLAM system to improve the accuracy and robustness, such as Dynamic Zero Velocity Update.

Author Contributions All authors contributed to the study's conception and design. Material preparation, data collection, and analysis were performed by Linqiu Gui, Chunnian Zeng, and Jie Luo. The first draft of the manuscript was written by Linqiu Gui. The paper was reviewed by Samuel Dauchert and Xiaofeng Wang. All authors commented on previous versions of the manuscript. All authors read and approved the final manuscript.

Funding The authors acknowledge the Chongqing Research Institute of Wuhan University of Technology project YF2021-16 for funding this work.

Code Availability Not available publicly.

Declarations

Competing Interests The authors have no relevant financial or non-financial interests to disclose.

Ethics Approval and Consent to Participate This study doesn't involve human or animal subjects.

Consent for Publication This study doesn't contain any individual person's data in any form. All the authors have agreed to publish this work.

References

1. Singandhupe, A., La, H.M.: A review of slam techniques and security in autonomous driving. In 2019 Third IEEE International Conference on Robotic Computing (IRC). IEEE, pp. 602–607 (2019)
2. Cadena, C., Carlone, L., Carrillo, H., Latif, Y., Scaramuzza, D., Neira, J., Reid, I., Leonard, J.J.: Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age. IEEE Trans. Robot. **32**(6), 1309–1332 (2016)
3. Shan, T., Englot, B., Meyers, D., Wang, W., Ratti, C., Daniela, R.: Lio-sam: Tightly-coupled lidar inertial odometry via smoothing and mapping. In IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, pp. 5135–5142 (2020)
4. Zhang, J., Singh, S.: Low-drift and real-time lidar odometry and mapping. Auton. Robot. **41**(2), 401–416 (2017)
5. Shan, T., Englot, B.: Lego-loam: Lightweight and ground-optimized lidar odometry and mapping on variable terrain. In IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, pp. 4758–4765 (2018)
6. Dube, R., Dugas, D., Stumm, E., Nieto, J., Siegwart, R., Cadena, C.: Segmatch: Segment based place recognition in 3d point clouds. In 2017 IEEE International Conference on Robotics and Automation (ICRA), pp. 5266–5272 (2017)
7. Pan, Y., Xiao, P., He, Y., Shao, Z., Li, Z.: Mulls: Versatile lidar slam via multi-metric linear least square. arXiv preprint arXiv:2102.03771 (2021)
8. Ye, H., Chen, Y., Liu, M.: Tightly coupled 3d lidar inertial odometry and mapping. In 2019 International Conference on Robotics and Automation (ICRA), pp. 3144–3150 (2019)
9. Besl, P., McKay, H.: A method for registration of 3-d shapes. IEEE Trans. Pattern Anal. Mach. Intell. **14**(2), 239–256 (1992)
10. Segal, A., Haehnel, D., Thrun, S.: Generalized-icp. In Robotics: Science and systems, vol. 2, no. 4. Seattle, WA, p. 435 (2009)
11. Yang, S., Zhu, X., Nian, X., Feng, L., Qu, X., Mal, T.: A robust pose graph approach for city scale lidar mapping. In 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 1175–1182 (2018)
12. Hening, S., Ippolito, C.A., Krishnakumar, K.S., Stepanyan, V., Teodorescu, M.: 3d lidar slam integration with gps/ins for uavs in urban gps-degraded environments. In AIAA Information Systems-AIAA Infotech @ Aerospace (2017)
13. Forster, C., Carlone, L., Dellaert, F., Scaramuzza, D.: On-manifold preintegration for real-time visual-inertial odometry. IEEE Trans. Robot. **33**(1), 1–21 (2016)
14. Qin, C., Ye, H., Pranata, C.E., Han, J., Zhang, S., Liu, M.: R-lins: A robocentric lidar-inertial state estimator for robust and efficient navigation. arXiv preprint arXiv:1907.02233 (2019)
15. Mur-Artal, R., Tardos, J.D.: Visual-inertial monocular slam with map reuse. In IEEE Robotics and Automation Letters **2**(2), 796–803 (2017)
16. Faessler, M., Fontana, F., Forster, C., Scaramuzza, D.: Automatic re-initialization and failure recovery for aggressive flight with a monocular vision-based quadrotor. In 2015 IEEE International Conference on Robotics and Automation (ICRA), pp. 1722–1729 (2015)
17. Qin, T., Li, P., Shen, S.: Vins-mono: A robust and versatile monocular visual-inertial state estimator. IEEE Trans. Robot. **34**(4), 1004–1020 (2018)
18. Wang, Z., Zhao, H., Qiu, S., Gao, Q.: Stance-phase detection for zupt-aided foot-mounted pedestrian navigation system. IEEE/ASME Trans. Mechatron. **20**(6), 3170–3181 (2015)
19. Norrdine, A., Kasmi, Z., Blankenbach, J.: Step detection for zupt-aided inertial pedestrian navigation system using foot-mounted permanent magnet. IEEE Sensors J. **16**(17), 6766–6773 (2016)
20. Kilic, C., Ohi, N., Gu, Y., Gross, J.N.: Slip-based autonomous zupt through gaussian process to improve planetary rover localization. IEEE Robot. Autom. Lett. **6**(3), 4782–4789 (2021)
21. Brossard, M., Barrau, A., Bonnabel, S.: Rins-w: Robust inertial navigation system on wheels. In 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, pp. 2068–2075 (2019)
22. Park, S.K., Suh, Y.S.: A zero velocity detection algorithm using inertial sensors for pedestrian navigation systems. Sensors **10**(10), 9163–9178 (2010)
23. Skog, I., Nilsson, J.O., Händel, P.: Evaluation of zero-velocity detectors for foot-mounted inertial navigation systems. In 2010 International Conference on Indoor Positioning and Indoor Navigation. IEEE, pp. 1–6 (2010)
24. Xiaofang, L., Yuliang, M., Ling, X., Jiabin, C., Chunlei, S.: Applications of zero-velocity detector and kalman filter in zero velocity update for inertial navigation system. On Proceedings of 2014 IEEE Chinese Guidance, Navigation and Control Conference. IEEE, pp. 1760–1763 (2014)
25. Ramanandan, A., Chen, A., Farrell, J.A.: Inertial navigation aiding by stationary updates. IEEE Trans. Intell. Transp. Syst. **13**(1), 235–248 (2011)
26. Dellaert, F., Kaess, M., et al.: Factor graphs for robot perception. Foundations and Trends® in Robotics **6**(1-2)1–139 (2017)
27. Skog, I., Handel, P., Nilsson, J.O., Rantakokko, J.: Zero-velocity detection-an algorithm evaluation. IEEE Trans. Biomed. Eng. **57**(11), 2657–2666 (2010)
28. Agarwal, S., Mierle, K., et al.: Ceres solver. http://ceres-solver.org
29. Kaess, M., Ranganathan, A., Dellaert, F.: isam: Incremental smoothing and mapping. IEEE Trans. Robot. **24**(6), 1365–1378 (2008)
30. Kim, G., Kim, A.: Scan context: Egocentric spatial descriptor for place recognition within 3d point cloud map. In 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, pp. 4802–4809 (2018)
31. Gao, W.: imu_utils. https://github.com/gaowenliang/imu_utils (2018)
32. Quigley, M., Conley, K., Gerkey, B., Faust, J., Foote, T., Leibs, J., Wheeler, R., Ng, A.Y.: Ros: an open-source robot operating system. In ICRA Workshop on Open Source Software, vol. 3, no. 3.2. Kobe, Japan, p. 5 (2009)

Linqiu Gui received his B.S. degree in Automation and M.S. in Control Science and Engineering from the Wuhan University of Technology, China, in 2014 and 2017, respectively. Currently, he is pursuing his PhD degree in Information and Communication Engineering at the Wuhan University of Technology, China. His research interests include Autonomous vehicles and navigation systems.

Chunnian Zeng is a Professor at the Wuhan University of Technology, China. Chunnian Zeng received his B.S. degree from the Huazhong University of Science and Technology, China, in 1982, and his M.S. degree from the Wuhan University of Technology, China, in 1987. He has been the vice-president of the Wuhan University of Technology and the director of the Center for Scientific and Technological Cooperation and Achievement Transformation.

Samuel Dauchert received his B.S. degree in Electrical Engineering from the University of South Carolina, Columbia in 2016. Currently, he is pursuing his PhD degree in Electrical Engineering at the University of South Carolina, Columbia. His research interests include robotics, navigation systems, and adaptive control.

Jie Luo received his B.S. degree in Automation, M.S. in Control Science and Engineering, and PhD in Information and Communication Engineering from the Wuhan University of Technology, China, in 2006, 2010, and 2013, respectively. He has been a Visiting Scholar at the University of South Carolina, USA, in 2018. He is currently an associate professor in the School of Automation at the Wuhan University of Technology. His research interests include Automatic control theory and application, Autonomous vehicles and embedded system.

Xiaofeng Wang received his B.S. degree in Applied Mathematics and M.S. in Operation Research and Control Theory from East China Normal University, China, in 2000 and 2003, respectively, and obtained his PhD degree in Electrical Engineering at the University of Notre Dame in 2009. After that, he worked as postdoctoral research associate at the University of Illinois at Urbana and Champaign. He is currently an associate professor in the Department of Electrical Engineering at the University of South Carolina, Columbia. His research interests include control theory, robotics, cyber-physical systems, and machine learning.