High-Precision SLAM Based on the Tight Coupling of Dual Lidar Inertial Odometry for Multi-Scene Applications

Kui Xiao 1*, Wentao Yu 1,*, Weirong Liu 2, Feng Qu 1 and Zhenyan Ma 1

1 College of Computer and Information Engineering, Central South University of Forestry and Technology, Changsha 410004, China; xiaokui@csuft.edu.cn (K.X.); ma@csuft.edu.cn (Z.M.)
2 School of Computer Science and Engineering, Central South University, Changsha 410083, China;

* Correspondence: wentaoyu@csuft.edu.cn; Tel.: +86-0731-134-6758-9376

Featured Application: This paper fuses different sensors to form a general high-precision SLAM framework for multi-scene applications. The algorithm framework in this paper can be extended to the fields of autonomous driving, robot navigation, and 3D reconstruction.

Abstract: Simultaneous Localization and Mapping (SLAM) is an essential feature in many applications of mobile vehicles. To solve the problem of poor positioning accuracy, single use of mapping scene, and unclear structural characteristics in indoor and outdoor SLAM, a new framework of tight coupling of dual lidar inertial odometry is proposed in this paper. Firstly, through external calibration and an adaptive timestamp synchronization algorithm, the horizontal and vertical lidar data are fused, which compensates for the narrow vertical field of view (FOV) of the lidar and makes the characteristics of vertical direction more complete in the mapping process. Secondly, the dual lidar data is tightly coupled with an Inertial Measurement Unit (IMU) to eliminate the motion distortion of the dual lidar odometry. Then, the value of the lidar odometry after correcting distortion and the pre-integrated value of IMU are used as constraints to establish a non-linear least-squares objective function. Joint optimization is then performed to obtain the best value of the IMU state values, which will be used to predict the state of IMU at the next time step. Finally, experimental results are presented to verify the effectiveness of the proposed method.

Keywords: simultaneous localization and mapping; dual lidar inertial odometry; IMU; time synchronization; tight coupling

1. Introduction

Simultaneous localization and mapping (SLAM) require building a map of an unknown environment by a mobile vehicle and simultaneously localizing the vehicle in such a map [1–3]. SLAM is essential for vehicles to fulfill many tasks, including vehicle rescue [4] and exploration [5]. The perception of the unknown external environment by various onboard sensors provides vital information for SLAM. Thus, integrating different sensors to develop a practical SLAM framework that can be applied in multiple scenes is essential.

Generally, both vision-based and lidar-based SLAM [6–10] are used. Although the vision-based SLAM can obtain high-precision positioning [11–13], the vision sensors are vulnerable to light change in the environment and cannot work in dark or untextured scenes. In comparison, lidar is not affected by light and can usually measure the angle and distance of obstacles with higher accuracy. Therefore, this paper focuses on the design of lidar-based SLAM to adapt the multi-scene applications.

In recent years, many different lidar-based SLAM schemes have been proposed. Among them, the Lidar Odometry and Mapping in Real-time (LOAM) method, where a single-line lidar and a motor are used to form a multiline lidar to realize low-drift and...
low-computation real-time positioning and mapping, has been studied extensively [14]. In the LOAM method, SLAM is divided into two parts: lidar odometry and lidar mapping. In the lidar odometry part, to reduce the computation, the plane smoothness of the lidar point cloud, which is utilized to distinguish the edge points, is calculated according to the curvature and then invalid point clouds are discarded to improve the feature of point cloud. After the point cloud information is obtained through feature extraction, a scan-to-scan method [15] is proposed to realize feature matching between frames, including edge points and planar points matching. To eliminate the motion distortion of lidar, linear interpolation is utilized. After the distortion is eliminated, the pose transformation of the lidar point cloud data of two adjacent frames is acquired to obtain the lidar odometry. Then, after accumulating a certain number of point cloud data, lidar mapping is performed through a map-to-map matching method [16]. Although LOAM can create a high-precision point cloud map, it cannot remove moving people or objects during feature extraction. Furthermore, in an environment with fewer feature points, it is easy for the lidar odometry to fail, resulting in positioning and mapping with worse accuracy.

To solve the above problems, a Lightweight and Ground-Optimized Lidar Odometry and Mapping on Variable Terrain (LeGO-LOAM) [17] is proposed. Its core comprises four modules: point cloud segmentation and denoising, feature extraction, lidar odometry, and lidar mapping. Firstly, the point cloud segmentation technology [18] solves the defects of moving people or objects in LOAM mapping and filters out noise. Then, feature extraction is applied to obtain planar and edge features from the segmented point cloud. The lidar odometry module uses the features to find out the optimal pose transform between two consecutive scans with the help of a two-step Levenberg–Marquardt (LM) optimization method [19]. The extracted features are further processed by a scan-to-map [20] matching method to obtain a global point cloud map in lidar mapping. LeGO-LOAM optimizes LOAM to a certain extent, but in large scenes (e.g., long corridors) or environments with few feature points, the low frequency of lidar and less characteristic information can lead to significant errors in positioning and mapping.

Some recent research has used low-cost IMU [21–23] to assist lidar for SLAM. The simplest way to integrate the IMU with lidar is loose coupling [24], in which IMU is regarded as an independent module to assist the lidar. The authors of [25] loosely couple IMU and optional GPS measurement with lidar through the Extended Kalman Filter (EKF) to improve computational efficiency and accuracy. However, the IMU error will continue to accumulate in the case of long distances, so the localization error is still increasing in such cases. To solve this problem, IMU and lidar are generally coupled via a tight coupling method that usually offers improved accuracy. A tightly coupled lidar inertial odometry and mapping framework (LIO-Mapping) is introduced in the literature [26]. It comprises the state optimization for the odometry and the refinement with rotational constraints. Results showed that this method outperformed the state-of-the-art lidar-only and loosely coupled methods. Since LIO-Mapping is designed to process all sensor measurements, real-time performance is not achieved. The authors of [27] proposed a tightly coupled Lidar Inertial Odometry via Smoothing and Mapping (LIO-SAM) based on LeGO-LOAM. LIO-SAM performs highly accurate, real-time mobile vehicle trajectory estimation and map building, suitable for multi-sensor fusion and global optimization. Although the tight coupling of lidar and IMU can improve the localization accuracy, lidar sensors have certain drawbacks. Conventional lidar can only be used to scan the environment in a narrow range of vertical angles and the point cloud feature information obtained is limited. For example, in an indoor corridor or staircase environment, lidar can receive the point cloud information of the sidewall, but only a tiny part of the point cloud information is obtained from the floor and ceiling. In such cases, the matched lidar features can easily cause ill-constrained pose estimation and incomplete mapping.

Aiming to broaden the FOV of lidar, lidars can be actuated in a number of ways. The periodic nodding and continuous rotation can be adopted and the resulting configurations can potentially enlarge the FOV. While the implementation of this method depends on
a complex mechanical structure, the localization accuracy is relatively low [28,29]. More recently, a new design for a 3D sensor system was introduced [30], involving construction from a 2D range scanner coupled with a passive linkage mechanism, such as a spring. However, the system requires reasonably continual excitation to induce motion of the sensor head. Therefore, the system is not considered appropriate for electric ground vehicles operating with infrequent accelerations on smooth terrain. In order to solve the above problems, some researchers have started investigating the use of multiple 3D lidars for better coverage of the environment. In [31], a high-precision lidar odometry system was proposed to achieve robust and real-time operation under challenging perceptual conditions. In this system, the two lidars are mounted at 180 degrees to each other to make up for the self-similar areas with low lidar observability. The authors of [32] proposed a system to achieve robust and simultaneous extrinsic calibration, odometry, and mapping for multiple lidars, while [33] proposed a scheme to combine multiple lidars with complementary FOV for feature-based lidar-inertia odometry and mapping. While the above methods can work well in single-scene applications, their adaptability to different environments was not considered, especially the environments that need to perceive height information.

This paper proposes a high-precision SLAM framework based on tight coupling of dual lidar inertial to overcome the above problems. In this framework, the horizontal lidar and the vertical lidar are integrated to broaden the FOV. Then the dual lidar and IMU are tightly coupled to improve the localization accuracy. In the outdoor environment, this paper introduces GPS measurements to further improve positioning accuracy. The main contributions of this paper include:

1. A general high-precision SLAM framework is provided by fusing different sensors. It can adapt to multi-scene applications, such as a corridor with fewer features, stairs with height, and complex outdoor environments.
2. The horizontal and vertical lidars are fused by external calibration and adaptive time synchronization algorithms to solve the narrow vertical FOV of the lidar.
3. To improve the positioning accuracy of SLAM in the environment with height information (e.g., stairs), the dual lidar odometry and IMU are tightly coupled. The dual lidar odometry measurement and IMU pre-integration are jointly optimized to obtain more accurate IMU state values, which will be used to eliminate the motion distortion of the dual lidar to improve the localization accuracy.
4. In addition, several practical experiments are carried out to verify the feasibility and effectiveness of the proposed method.

2. Prerequisites

2.1. IMU State Prediction and Pre-Integration

Usually, the frequency of IMU is much higher than lidar. IMU can obtain the acceleration and angular velocity at each time step, which can be used to predict the next state of IMU through integration.

In general, the IMU states at time $t$ can be modeled as:

$$X_W^t = [p_t, v_t, R_t, b_t]$$

where $X_W^t$ represents the state in the world frame $W$ at time $t$; $p_t$, $v_t$, and $R_t$ represent the position, velocity, and rotation matrix, respectively, at time $t$; and $\hat{a}_t$ and $\hat{\omega}_t$ are the acceleration and angular velocity of the raw IMU measurements ($\hat{a}_t$ and $\hat{\omega}_t$ are affected by a slowly varying bias, $b_t$).

The next state, $X_W^{t+\Delta t}$, is predicted through the integration of IMU, where $\Delta t$ is the interval between two consecutive IMU measurements. The state prediction value of the
IMU is then used to infer the motion of the vehicle. The velocity, position, and rotation of the vehicle at time \( t + \Delta t \) can be computed by Equation (2) [27]:

\[
\begin{align*}
    v_{t+\Delta t} &= v_t + g\Delta t + R_t(\Delta t - b^a_t - n^a_t)\Delta t \\
    p_{t+\Delta t} &= p_t + v_t\Delta t + \frac{1}{2}g\Delta t^2 + \frac{1}{2}R_t(\Delta t - b^a_t - n^a_t)\Delta t^2 \\
    q_{t+\Delta t} &= q_t \otimes \left[ \frac{1}{2}\Delta t(\omega_{\Delta t} - b^{\omega}_{\Delta t} - n^{\omega}_{\Delta t}) \right]
\end{align*}
\]  

(2)

where \( \mathbf{g} \) is gravitational acceleration; \( \mathbf{b}^a_t \) is the varying bias of the acceleration \( \mathbf{a}_t \); and \( \mathbf{n}^a_t \) is the white noise of the acceleration \( \mathbf{a}_t \). The quaternion \( \mathbf{q}_t \) under Hamilton notation (which corresponds to \( \mathbf{R}_t \)) is used; \( \otimes \) is used for the multiplication of two quaternions; \( \mathbf{b}^{\omega}_t \) is the varying bias of the angular velocity \( \mathbf{\omega}_t \); and \( \mathbf{n}^{\omega}_t \) is the white noise of the angular velocity \( \mathbf{\omega}_t \).

The motion state of IMU between the \( i \)-th and the \( j \)-th time steps can be represented by the IMU pre-integrated measurement value \( \Delta \mathbf{p}_{ij}, \Delta \mathbf{v}_{ij}, \Delta \mathbf{q}_{ij} \), which can be computed by:

\[
\begin{align*}
    \Delta \mathbf{v}_{ij} &= \mathbf{R}_i^T(\mathbf{v}_j - \mathbf{v}_i - \mathbf{g}\Delta t_{ij}) \\
    \Delta \mathbf{p}_{ij} &= \mathbf{R}_i^T(\mathbf{p}_j - \mathbf{p}_i - \mathbf{v}_i\Delta t_{ij} - \frac{1}{2}\mathbf{g}\Delta t_{ij}^2) \\
    \Delta \mathbf{q}_{ij} &= \mathbf{q}_i^{-1} \otimes \mathbf{q}_j = \prod_{k=1}^{j-1} \left[ \frac{1}{2}\Delta t_{ij} (\mathbf{\omega}_k - b^{\omega}_k - n^{\omega}_k) \right]
\end{align*}
\]  

(3)

where \( \Delta \mathbf{p}_{ij}, \Delta \mathbf{v}_{ij}, \Delta \mathbf{q}_{ij} \) is the position, velocity, and rotation matrix of the IMU, respectively, which has the covariance \( \mathbf{C}^j_i \), in the error-state model. Due to space limitations, we invite readers to refer to the descriptions in [26,34] to understand the detailed derivation of IMU pre-integration.

2.2. Segmentation and Feature Extraction

Before extracting features, to filter out the noise, the point cloud is segmented to reduce noise interference. \( \mathbf{D}_t = \{ \mathbf{d}_1, \mathbf{d}_2, \ldots, \mathbf{d}_n \} \) is the point cloud information acquired at time \( t \); \( \mathbf{d}_\tau \) is a point in \( \mathbf{D}_t; \mathbf{D}_t \) is projected onto a range image; the point \( \mathbf{d}_\tau \) represents a pixel of the image; and the pixel value \( \mathbf{r}_\tau \) represents the Euclidean distance from the point \( \mathbf{d}_\tau \) to the lidar. An image-based segmentation method [17] is applied to the range image to group points into many clusters. Points of the same category have the same label (the ground is a particular category). Features are then extracted from ground points and segmented points, with the corresponding edge points and plane points obtained by calculating the curvature \( \mathbf{E} \) of each point. To evenly extract features from all directions, the range image is divided horizontally into several equal sub-images. Then, the points are sorted in each row of the sub-image based on their curvature values, \( \mathbf{E} \). The curvature \( \mathbf{E} \) is computed by Equation (4).

\[
\mathbf{E} = \frac{1}{|\mathbf{S}| \cdot \| \mathbf{r}_\tau \|} \sum_{\lambda \in \mathbf{S}, \lambda \neq \tau} (\mathbf{r}_\lambda - \mathbf{r}_\tau)
\]  

(4)

where \( \mathbf{S} \) is the set of continuous points \( \mathbf{d}_\tau \) from the same row of the range image; \( \lambda \) is a point in the set \( \mathbf{S} \); \( \mathbf{E}_{th} \) is a threshold which is set to distinguish different types of features; the points are called with \( \mathbf{E} > \mathbf{E}_{th} \) edge features; and the points are called with \( \mathbf{E} < \mathbf{E}_{th} \) planar features.

3. Multi-Sensor Fusion

3.1. Hardware System Description

The configuration of our system is shown in Figure 1, with the left model obtained through Unigraphics NX and the designed equipment in our paper shown on the right. The designed equipment is a scanning arm composed of various sensors that can be installed on a backpack or a vehicle. The sensor suite used in the scanning arm includes two 10 HZ Velodyne VLP-16 lidars, a 200 HZ YIS510A IMU, and a 5 HZ HY-269 GPS. The panoramic
camera prepares for the follow-up research. The lidar has a horizontal FOV of 360° with 0.4° resolution, a vertical FOV 15° with 2° resolution, and an accuracy of ±3 cm. The Roll/Pitch of IMU accuracy is 0.25°, HY-269 GPS is a small-volume positioning and directional receiver, and the positioning accuracy of GPS is 1.2 m.

![Figure 1](image1.jpg)

**Figure 1.** A CAD (Computer Aided Design) model of the scanning arm is shown on the left, with a photograph of the scanning arm shown on the right.

### 3.2. System Overview

In this paper, horizontal and vertical lidars are fused to make up for the shortcomings of lidar’s narrow FOV. When indoors, accurate positioning information is obtained through the tight coupling of dual lidar inertial odometry. When outdoors, GPS measurements are added, providing a relatively accurate initial position for positioning, thereby further improving outdoor positioning accuracy.

Figure 2 provides a brief overview of our proposed framework. Firstly, after obtaining the IMU data, the IMU state prediction and pre-integration are updated, as detailed in Section 2.1. IMU pre-integration is used to construct a factor graph constraint, which will be used for joint optimization. Secondly, the dual lidar point clouds are obtained through external calibration and an adaptive timestamp synchronization algorithm, as detailed in Section 3.3. Since lidar has continuous measurement, the lidar measurement is obtained in continuous motion and it is almost certain that motion distortion will occur. When dual lidar point cloud is received, deskewing is applied on the dual lidar raw point cloud to obtain deskewed dual lidar point cloud [14]. To be ground-optimized and reduce the amount of calculation, point cloud segmentation is applied to filter out noise (e.g., moving people or objects) and the planar points and edge points are distinguished by calculating the curvature of the lidar point, as detailed in Section 2.2. After feature extraction, the dual lidar point cloud is registered to construct the lidar odometer factor. Then, the sliding window factor graph optimization method of local finite frame is used, with the IMU pre-integration factor and lidar odometer factor jointly optimized to realize the mutual parameter optimization of the dual lidar and IMU. Joint optimization is taken to obtain a maximum a posteriori (MAP) estimation of the IMU states, as detailed in Section 3.4, avoiding the drift from IMU propagation. To further improve outdoor positioning accuracy, GPS is introduced in this framework. When GPS is loosely coupled with IMU through an Unscented Kalman Filter (UKF), GPS measurements are used to further improve the positioning accuracy of IMU, with the optimized IMU state then used for the state estimation of IMU at the next time step. GPS measurements are added, providing a relatively accurate initial position for localization, thereby further improving outdoor localization accuracy [25]. However, the
drift of lidar inertial odometry grows very slowly; in practice, GPS measurements are used when the estimated position covariance is larger than the received GPS position covariance. Finally, the output of the figure is the global map and localization.

3.3. Fusion of Horizontal Lidar and Vertical Lidar

To accurately fuse the horizontal and vertical lidar data, their individual coordinate systems must be transformed into a unified coordinate system, which is termed “frame coordinate system”. The horizontal lidar coordinate system is used as the frame coordinate system; the vertical lidar coordinate system is registered in this frame coordinate system through external parameter calibration, with the external parameters obtained by the CAD model of the scanning arm. This paper adopts the adaptive time synchronization algorithm to ensure that the timestamps of the horizontal and vertical lidars can be simultaneously output. The fusion process of dual lidar is divided into two parts: (1) external parameter calibration of the horizontal and vertical lidars and (2) the adaptive time synchronization algorithm. Both parts are described as follows.

3.3.1. External Parameter Calibration of Horizontal Lidar and Vertical Lidar

Assuming that \( L_1 \) is the horizontal lidar coordinate system and \( L_2 \) is the vertical lidar coordinate system, the horizontal lidar coordinate system is used as the frame coordinate system and the vertical lidar coordinate system \( L_2 \) is registered in this frame coordinate system \( L_1 \) through the rotation matrix \( \mathbb{R}^{L_1}_{L_2} \) and translation matrix \( H^{L_1}_{L_2} \), which can be computed by Equation (5):
\[
L_1 = \mathbb{R}^{L_1}_{L_2} \times L_2 + H^{L_1}_{L_2}
\]
where \( \mathbb{R}^{L_1}_{L_2} \) is the rotation matrix and \( H^{L_1}_{L_2} \) is the translation matrix which can be obtained by external parameters. From the hardware structure of the scanning arm, the vertical lidar coordinate system first rotates around the \( y \)-axis and then translates to the horizontal lidar coordinate system, where
\[
\mathbb{R}^{L_1}_{L_2} = \begin{bmatrix}
\cos \theta & 0 & \sin \theta \\
0 & 1 & 0 \\
-\sin \theta & 0 & \cos \theta
\end{bmatrix}
\]

Figure 2. An overview of our proposed framework.
where the angle of the rotation matrix \( \theta = -77.94^\circ \) and the translation matrix \( H_{L_2}^{L_1} = (x, y, z) \), with \( x = -0.177 \) m and \( z = -0.213 \) m.

The point cloud of the lidar after external parameter calibration in the frame coordinate system is shown in Figure 3, with the point clouds of the vertical and horizontal lidars not affecting each other.

![Figure 3. Dual lidar point clouds' information after external parameter calibration.](image)

3.3.2. The Adaptive Time Synchronization Algorithm

In this paper, the lidar time synchronization is achieved through hardware time synchronization and GNSS is used as the master clock. However, there is a time offset between the two lidars. As the time offset is constant, an adaptive timestamp synchronization algorithm is used to solve it. The adaptive time synchronization algorithm is used to match its timestamp and realize the simultaneous localization and mapping of the lidars. The timestamps of the dual lidar is shown in Figure 4, where lidar_201/scan represents the timestamp of the horizontal lidar and lidar_202/scan represents the timestamp of the vertical lidar.

![Figure 4. Timestamps of dual lidar.](image)

The output of the adaptive time synchronization algorithm only depends on the timestamp, not on the arrival time of dual lidar point cloud messages. It means that the adaptive time synchronization algorithm can be safely used on dual lidar point cloud messages that have suffered arbitrary processing delays. As shown in Figure 5, time goes from left to right, with the first row representing the horizontal lidar timestamp and the second row representing the vertical lidar timestamp. Each dot represents the lidar point cloud with the timestamp. The blue dot represents the pivot of the lidar point clouds, with the broken line linking the dual lidar point clouds in a set. Suppose the horizontal lidar point clouds’ queue is \( PL_P \) and the vertical lidar point clouds’ queue is \( PV_P \), with lidar point clouds inserted in a topic-specific queue (\( PL_P \) or \( PV_P \)) as they arrive. Once each topic-specific queue contains at least one message, the latest message is found at the head of the queues, known as the pivot. Assume that the queue with pivots is \( PL_P \), which matches \( PV_P \), being the smallest difference between the timestamps, \( T \). The two queues are combined into a set; finally, this set is synchronous output.
3.4. Tight Coupling of Dual Lidar and IMU

Although the fusion of dual lidar makes up for the shortcomings of lidar’s narrow FOV, the localization accuracy is improved. However, lidars mounted on moving vehicles suffer from motion distortion, which directly affects the localization accuracy. IMU can accurately measure the three-axis acceleration and three-axis angular velocity of moving vehicles, and provide a priori information for lidar odometry. However, in the case of long or short distances, the IMU error will continue to accumulate, which directly affects positioning accuracy. In this paper, the tight coupling of dual lidar and IMU is proposed to solve the above problems. The process is divided into two parts: (1) external parameter calibration of horizontal lidar and IMU and time synchronization, and (2) joint optimization, which are described as follows:

3.4.1. External Parameter Calibration of Horizontal Lidar and IMU and Time Synchronization

Similar to Section 3.3, the external parameters of the lidar and IMU are also obtained based on the CAD model. To accurately fuse data from the horizontal lidar, the vertical lidar, and the IMU, the IMU coordinate system \(I\) is also registered in this frame coordinate system \(L_1\) through the rotation matrix \(R^{L_1}_I\) and the translation matrix \(H^{L_1}_I\). It can be seen from the structure of the scanning arm in Figure 1 that the IMU is located directly under the horizontal lidar. Therefore, the IMU coordinate system can be registered in the horizontal lidar coordinate system only by translation, where

\[
R^{L_1}_I = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}
\]  

(7)

where the translation matrix \(H^{L_1}_I = [-0.6, 2.66, 0, -0.125]\).

In this paper, YIS510A IMU is used, which supports trigger correction of its internal clock by PPS signal. Lidar-IMU time synchronization is achieved through hardware time synchronization, with GNSS used as the master clock. GNSS can output PPS signals to unify the clock source of lidar-IMU and achieve time synchronization.

3.4.2. Joint Optimization

In this paper, a fixed-lag smoother and marginalization are introduced to obtain the optimal state. The fixed-lag smoother maintains a certain amount of IMU state in the sliding window, and the sliding window can effectively control the amount of calculation. When the new state enters the sliding window, the past state is marginalized. The state variable to be estimated for the whole window is \(X = [X^W_\eta, ..., X^W_{\eta + \kappa}, Z^L_{\eta + 1}]\), where \(X^W_\eta\) is the state of IMU at the starting point \(\eta\) of the sliding window; \(X^W_{\kappa}\) is the state of IMU at the end of the sliding window; and \(Z^L_{\kappa} = [R^{L_1}_{\kappa}, H^{L_1}_{\kappa}]\) is the external parameter between lidar and IMU. Then, the following cost function with a Mahalanobis norm is minimized to obtain the MAP estimation of the states \(X\),

\[
X = \min_X \frac{1}{2} \left\{ u_o(X)^2 + \sum_{x \in Q_{\eta+1, \kappa}} u_x(x, X)^2 + \sum_{\varsigma \in \{\eta, ..., \kappa - 1\}} \sum_{\varsigma + 1} u_\varsigma(Z^L_{\varsigma + 1}, X)^2 \right\}
\]  

(8)
where \( u_o(X) \) is the prior information from marginalization [26]. \( u_\phi(\chi, X) \) is the residual of the relative lidar constraints that can be represented as point-to-plane distance [26], where \( \chi \in Q_\beta \) is the residual for each relative lidar measurement with the previous correspondences. \( \beta \in \{ \eta + 1, \ldots, \kappa \} \), \( \eta + 1 \) and \( \kappa \) are the timestamps of the lidar sweep next to the starting one and the current lidar sweep in the window, with the covariance matrix \( C_{L, \beta}^X \) determined by the lidar accuracy [35]. \( u_\gamma(Z_{\xi+1}^c, X) \) is the residual of the IMU constraints, where

\[
\begin{align*}
    u_\gamma(Z_{\xi+1}^c, X) =
    & \begin{bmatrix}
        R_{\xi}^T \left( p_{\xi+1} - p_{\xi} - v_{\xi}\Delta t - \frac{1}{2}g\Delta t^2 \right) - \Delta p_{\xi+1}
        \\
        R_{\xi}^T \left( v_{\xi+1} - v_{\xi} - g\Delta t_{\xi+1} \right) - \Delta v_{\xi+1}
        \\
        2 \left[ (\Delta q_{\xi+1}^{-1} \otimes q_{\xi}^{-1} \otimes q_{\xi+1})_{xyz} \right]
        \\
        b_{\xi+1}^a - b_{\xi}^a
        \\
        b_{\xi+1}^w - b_{\xi}^w
    \end{bmatrix}
\end{align*}
\]

\( u_\gamma(Z_{\xi+1}^c, X) \) can be obtained by IMU state prediction (Equation (2)) and IMU pre-integration (Equation (3)), and \( \left[ (\Delta q_{\xi+1}^{-1} \otimes q_{\xi}^{-1} \otimes q_{\xi+1})_{xyz} \right] \) stands for the vector part of a quaternion. With the continuous-time linearized propagation of the error states and the IMU noise parameters, the covariances \( C_{L, \xi} \) of the pre-integration measurements and biases can be estimated. The cost function, in the form of a non-linear least-square, can be solved by the Gauss–Newton algorithm. Ceres Solver [36] is used for solving this nonlinear problem.

It can be seen from Section 3.2 that the new states \( X \) are obtained by the joint optimization, which is used as the next state of the IMU, avoiding the drift from IMU propagation. The state of the IMU is applied to deskewing, thereby eliminating the motion distortion of the dual lidar.

4. Experiment

4.1. Data Acquisition Equipment

To verify the performance of the proposed high-precision SLAM framework based on the tight coupling of Dual Lidar Inertial Odometry (HSDLIO) for multi-scene applications, this paper describes a series of experiments to qualitatively and quantitatively analyze our proposed framework. The device runs on an industrial computer with a processor of Intel i7-9700 through the port and network port, and the system configured of the industrial computer is ubuntu16.04. The scanning arm is installed on the backpack, with the industrial computer and battery installed in the backpack. The small display screen of the backpack shows the localization and mapping in real-time, with the backpack equipment shown in Figure 6.

Data of three representative environments are collected, including the featureless corridor, stairs with height, and complex outdoor environments. To illustrate the effectiveness of the HSDLIO algorithm, LeGO-LOAM and LIO-SAM algorithms are compared with it.
4.2. Indoor Experiment 1

Indoor experiment 1 was carried out in a corridor with few feature points. Since there was no GPS signal indoors, SLAM was mainly carried out through the tight coupling of dual lidar odometry and IMU.

To highlight the performance of the HSDLIO algorithm, LeGO-LOAM and LIO-SAM algorithms are compared with it. Figure 7a shows the environment map of the corridor, while Figure 7b–d shows the overall mapping of LeGO-LOAM, LIO-SAM, and HSDLIO algorithms in the corridor scene, with the white points representing backpack motion trajectories. Compared with the HSDLIO algorithm in Figure 7d, the overall mapping of LeGO-LOAM (Figure 7b) has a significant deviation and the structure is unclear. Figure 8a,b shows that the mapping of LeGO-LOAM is incomplete because the lidar has a considerable drift in a scene with fewer feature points. The mapping results of LIO-SAM (Figure 7c) is close to the real environment, with its structure complete. The main reason is that LIO-SAM tightly couples the horizontal lidar and IMU to make up for the lidar drift. However, Figure 9a,b indicates the mapping result of HSDLIO is more abundant than LIO-SAM in the top surface and ground information. The reason is that HSDLIO fuses horizontal and vertical lidars so that its angle of FOV in both horizontal and vertical directions is 360°. The dual lidars and IMU are tightly coupled to make up for the lidar drift.

Figure 7. (a) Corridor scene for experiment 1. (b) Mapping results of LeGO-LOAM in the corridor scene, with the mapping of LeGO-LOAM having a significant deviation and the structure incomplete. (c) Mapping results of LIO-SAM in the corridor scene, with LIO-SAM having a complete mapping structure but lacking information on the top and ground. (d) Mapping results of HSDLIO in the corridor scene, with HSDLIO having a complete mapping structure.
Figure 8. Comparison of mapping details at the corner of the corridor, where (a) the mapping of LeGO-LOAM has a significant deviation and (b) the mapping of the HSDLIO structure is clearer and more complete than LeGO-LOAM.

Figure 9. Comparison of mapping details at the top of the corridor, where (a) the mapping of LIO-SAM lacks top information and (b) the mapping of HSDLIO shows finer structural details of the environment.

In the long corridor scene, compared with LeGO-LOAM and LIO-SAM, the structure of HSDLIO mapping is more complete and the point cloud information on the ground and top surface is richer. To verify the positioning accuracy of the HSDLIO, the trajectories of LeGO-LOAM, LIO-SAM, and HSDLIO are compared in Figure 10. It can be observed in Figure 10 that the LeGO-LOAM trajectory has drifted. The main reason is that in the LeGO-LOAM algorithm, the lidar odometry error continues to increase in a long corridor environment with fewer feature points. Both LIO-SAM and HSDLIO use the tight coupling of IMU and lidar for positioning, which improves positioning accuracy. HSDLIO further improves the accuracy of positioning through the tight coupling of dual lidars and IMU.

To further improve the positioning accuracy of HSDLIO, the relative translational error (RTE)—the distance from the start point to the end point—is introduced. In all experiments, the data collection started and ended at the same point; when the positioning accuracy of
the device is very high, the start and end point will coincide. The RTE, when the backpack returns to the start point, is shown in Table 1. The RTE can be computed by Equation (10):

$$RTE = \sqrt{x_o^2 + y_o^2 + z_o^2}$$ (10)

where $x_o = x_e - x_o$, $y_o = y_e - y_o$, $z_o = z_e - z_o$, $x_o$, $y_o$, and $z_o$ are the coordinates of the starting point and $x_e$, $y_e$, and $z_e$ are the coordinates of the end point. The result of LeGO-LOAM is not shown because its trajectory has severely shifted. Table 1 shows that the error of HSDLIO in the $x$ and $y$ direction is similar to LIO-SAM, but the accuracy in the $z$ direction is higher than LIO-SAM and the RTE of HSDLIO is smaller than LIO-SAM.

![Figure 10. Comparison of LeGO-LOAM, LIO-SAM, and HSDLIO trajectories. The red line is the trajectory of HSDLIO, the blue line is the trajectory of LIO-SAM, and the black line is the trajectory of LeGO-LOAM, with the latter having drifted.](image)

Table 1. Relative translational error when the backpack returns to the starting point (meters).

| Algorithm | LeGO-LOAM | LIO-SAM | HSDLIO |
|-----------|-----------|---------|--------|
| $x_o$     | Fail      | 0.079   | 0.077  |
| $y_o$     | Fail      | 0.092   | 0.090  |
| $z_o$     | Fail      | 0.121   | 0.001  |
| RTE       | Fail      | 0.171   | 0.118  |

4.3. Indoor Experiment 2

Indoor experiment 2 aimed to show the effectiveness of HSDLIO in the indoor environment with a certain height information. In this experiment, the backpack was carried up and down four flights of stairs. Figure 11a shows the stairs scene, while Figure 11b–d shows the overall mapping of LeGO-LOAM, LIO-SAM, and HSDLIO algorithms. The white points are backpack localization trajectories. LeGO-LOAM (Figure 11b) cannot obtain the structure of the stairs. Because the vertical FOV of the horizontal lidar is relatively narrow, the mapping result of LIO-SAM (Figure 11c) does not show the height information. With the help of the vertical lidar and the tight coupling of the IMU, the mapping of HSDLIO (Figure 11d) is the most complete and accurate, which is closer to the real stairs’ scene.

Figure 12 shows the trajectories obtained by LeGO-LOAM, LIO-SAM, and HSDLIO. The red, black, and green lines represent the trajectories of HSDLIO, LeGO-LOAM, and LIO-SAM, respectively, with the trajectory of HSDLIO approximating the real trajectory. LeGO-LOAM has no IMU correction, which causes the drift of the trajectory. Although the trajectory of LIO-SAM does not drift, due to the narrow vertical FOV of the horizontal lidar, a significant error occurs in the $z$ direction information. Comparing the height
information of LIO-SAM and HSDLIO in the z direction in Figure 13, it can be seen that when going up and down four floors of stairs, the localization results of LIO-SAM provide no height information and also indicate a great drift, while HSDLIO provides accurate height information. Therefore, the tight coupling of dual lidars and IMU makes the advantages of HSDLIO clearly apparent. Table 2 shows that the error of HSDLIO in the x and y direction is smaller than LIO-SAM, but the accuracy in the z direction is much higher than LIO-SAM and the RTE of HSDLIO is much smaller than LIO-SAM.

![Stairs scene for experiment 2](image)

**Figure 11.** (a) Stairs scene for experiment 2. (b) Mapping results of LeGO-LOAM in the stairs scene, with LeGO-LOAM’s mapping result having failed. (c) Mapping results of LIO-SAM in the stairs scene, with LIO-SAM’s mapping results lacking height information. (d) Mapping results of HSDLIO in the stairs scene, with the mapping of HSDLIO the most complete and accurate.

![Comparison of trajectories](image)

**Figure 12.** Comparison of LeGO-LOAM, LIO-SAM, and HSDLIO trajectories. The red line is the trajectory of HSDLIO, the blue line is the trajectory of LIO-SAM, and the black line is the trajectory of LeGO-LOAM. While LeGO-LOAM’s trajectory drifted and LIO-SAM’s trajectory produced cumulative errors in the z direction, the trajectory of HSDLIO approximated the real trajectory.

### 4.4. Outdoor Experiment

The outdoor experiment was carried out on a city street. This scene has rich feature points, so the drift of lidar odometry grows very slowly. The localization accuracy of LeGO-LOAM, LIO-SAM, and HSDLIO have very little difference, but the mapping of HSDLIO can provide finer structural details.
Figure 13. Comparison of the height information of LIO-SAM and HSDLIO in the $z$ direction. The trajectory of HSDLIO provides height information from going up and down four floors of stairs, while LIO-SAM failed to position in the $z$ direction.

Table 2. Relative translational error when the backpack returns to the starting point (meters).

| Algorithm | LeGO-LOAM | LIO-SAM | HSDLIO |
|-----------|-----------|---------|--------|
| $x_o$     | Fail      | 0.033   | 0.011  |
| $y_o$     | Fail      | 0.036   | 0.024  |
| $z_o$     | Fail      | 0.062   | 0.004  |
| RTE       | Fail      | 0.078   | 0.026  |

The trajectories obtained by LeGO-LOAM, LIO-SAM, and HSDLIO are shown in Figure 14. Because there is more feature information outdoors, the drift of lidar odometry grows very slowly. LeGO-LOAM can obtain relatively accurate localization only through horizontal lidar. In LIO-SAM and HSDLIO, GPS measurement is introduced by loosely coupling with IMU through UKF, which can provide relatively accurate initial positioning. The localization accuracy of HSDLIO and LIO-SAM is higher than LeGO-LOAM. The available data in Table 3 further supports the performance of the HSDLIO algorithm.

Figure 14. Comparison of LeGO-LOAM, LIO-SAM, and HSDLIO trajectories. The red dashed line is the trajectory of HSDLIO, the blue dashed line is the trajectory of LIO-SAM, and the black dashed line is the trajectory of LeGO-LOAM.
Table 3. Relative translational error when the backpack returns to the starting point (meters).

| Algorithm | LeGO-LOAM | LIO-SAM | HSDLIO |
|-----------|-----------|---------|--------|
| RTE       | 0.082     | 0.036   | 0.031  |

The 3D city street scene is shown in Figure 15a. In Figure 15a–d, the marks in the white circle represent trees and the marks in the yellow circle represent the building structure. Figure 15b–d shows the overall mapping of LeGO-LOAM, LIO-SAM, and HSDLIO algorithms in the city street scene, with the white points being their localization trajectories. Figure 15b,c shows that the overall effect of the mapping lacks building height information, while the structure of buildings and trees is blurred. The mapping details are analyzed at the marker, compared with the mapping results of LeGO-LOAM and LIO-SAM (Figure 15b,c), while the HSDLIO mapping result is shown in Figure 15d. It can be seen in Figure 15d that the structure of the trees and buildings is complete and has finer details.

![Figure 15.](image)

To further demonstrate the advantages of HSDLIO mapping, the mapping details of the building are shown in Figure 16, with the mapping results of LeGO-LOAM and LIO-SAM lacking the structural details and height information of the building. However, the mapping result of HSDLIO show finer structural details and more height information of the building.
Figure 16. Comparison of mapping details in the city street scene showing that (a) the mapping of LeGO-LOAM lacks structural details of the building, (b) the mapping of LIO-SAM lacks the height of the building structure, and (c) the mapping of HSDLIO provides finer structural details of the building.

5. Conclusions

This paper proposes a high-precision SLAM framework for multi-scene applications. In this framework, dual lidars are fused to make up for the shortcomings of lidars’ narrow FOV and hence improve the completeness of mapping. Meanwhile, dual lidars and IMU are tightly coupled to improve the localization accuracy. Extensive experiments were carried out and the results showed that compared with the commonly used LeGO-LOAM and LIO-SAM methods, our proposed method can produce more precise localization and more accurate mapping results with more details.

Author Contributions: Conceptualization, K.X.; methodology, K.X. and W.Y.; software, K.X., W.Y., W.L. and F.Q.; validation, W.Y. and Z.M.; formal analysis, K.X. and W.L.; investigation, W.L. and K.X.; resources, K.X. and W.Y.; data curation, K.X. and W.Y.; writing—original draft, K.X.; writing—review and editing, K.X. and W.Y.; visualization, K.X.; supervision, K.X. and Z.M.; project administration, K.X. and W.Y.; funding acquisition, K.X. and W.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation (grant nos. 61602529 and 61672539). This work was also supported by Hunan Key Laboratory of Intelligent Logistics Technology (2019TP1015) and Scientific Research Project of Hunan Education Department (No. I7C1650).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.
Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Dissanayake, M.W.M.G.; Newman, P.; Clark, S.; Durrant-Whyte, H.F.; Csorba, M. A solution to the simultaneous localization and map building (SLAM) problem. *IEEE Trans. Robot. Autom.* 2001, 17, 229–241. [CrossRef]

2. Durrant-Whyte, H.; Bailey, T. Simultaneous localization and mapping: Part I. *IEEE Robot. Autom. Mag.* 2006, 13, 99–110. [CrossRef]

3. Khairuddin, A.R.; Talib, M.S.; Haron, H. Review on Simultaneous Localization and Mapping (SLAM). In *Proceedings of the 2006 IEEE International Conference on Robotics and Automation*, Barcelona, Spain, 18–22 April 2005; pp. 661–666.

4. Belter, D.; Nowicki, M.; Skrzypczy’Nski, P. Lightweight RGB-D SLAM System for Search and Rescue Robots. In *Progress in Automation, Robotics and Measuring Techniques*; Springer: Cham, Switzerland, 2015; Volume 2, pp. 11–21.

5. Sim, R.; Roy, N. Global A-Optimal Robot Exploration in Slam. In *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, Barcelona, Spain, 18–22 April 2005; pp. 661–666.

6. Cheng, Z.; Wang, G. Real-Time RGB-D SLAM with Points and Lines. In *Proceedings of the 2018 2nd IEEE Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC)*, Xi’an, China, 25–27 May 2018; pp. 119–122.

7. Liu, T.; Zhang, X.; Wei, Z.; Yuan, Z. A robust fusion method for RGB-D SLAM. In *Proceedings of the 2013 Chinese Automation Congress*, Changsha, China, 7–8 November 2013; pp. 474–481.

8. Hess, W.; Kohler, D.; Rapp, H.; Andor, D. Real-Time Loop Closure in 2D LIDAR SLAM. In *Proceedings of the 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Macau, China, 3–8 November 2013; pp. 7750–7755.

9. Wang, Z.; Zhang, J.; Chen, S. Robust High Accuracy Visual-Inertial-Lidar SLAM System. In *Proceedings of the 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Macau, China, 3–8 November 2019; pp. 6636–6641.

10. Mur-Artal, R.; Tardós, J.D. ORB-SLAM2: An Open-Source SLAM System for Monocular, Stereo, and RGB-D Cameras. *IEEE Trans. Robot.* 2017, 33, 1255–1262. [CrossRef]

11. Zhou, Y.; Li, H.; Kneip, L. Canny-VO: Visual Odometry with RGB-D Cameras Based on Geometric 3-D–2-D Edge Alignment. *IEEE Trans. Robot.* 2019, 35, 184–199. [CrossRef]

12. Hu, G.; Huang, S.; Zhao, L.; Alempijevic, A.; Dissanayake, G. A robust RGB-D SLAM algorithm. In *Proceedings of the 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Vilamoura-Algarve, Portugal, 7–12 October 2012; pp. 1714–1719.

13. Zhang, J.; Singh, S. Loam: Lidar odometry and mapping in real-time. In *Proceedings of the Robotics: Science and Systems X*, Berkeley, CA, USA, 12–16 July 2014.

14. Wang, Z.; Zhang, J.; Chen, S. Robust High Accuracy Visual-Inertial-Lidar SLAM System. In *Proceedings of the 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Macau, China, 3–8 November 2019; pp. 6636–6641.

15. Tsardoulias, E.; Petrou, L. Critical rays scan match SLAM. *J. Intell. Robot. Syst.* 2013, 72, 441–462. [CrossRef]

16. Jiang, B.; Zhu, Y.; Liu, M. A Triangle Feature Based Map-to-map Matching and Loop Closure for 2D Graph SLAM. In *Proceedings of the 2019 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, Dali, China, 6–8 December 2019; pp. 2719–2725.

17. Shan, T.; Englot, B.J. Lego-loam: Lightweight and ground-optimized lidar odometry and mapping on variable terrain. In *Proceedings of the 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Madrid, Spain, 1–5 October 2018; pp. 4758–4765.

18. Bogoslavsvkyi, I.; Stachniss, C. Fast range image-based segmentation of sparse 3D lidar scans for online operation. In *Proceedings of the 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Daejeon, Korea, 9–14 October 2016; pp. 163–169.

19. Ranganathan, A. The levenberg-marquardt algorithm. *Tutoral LM Algorithm* 2004, 11, 101–110.

20. Torres-Torriti, M.; Guesalaga, A. Scan-to-map matching using the Hausdorff distance for robust mobile robot localization. In *Proceedings of the 2008 IEEE International Conference on Robotics and Automation (ICRA)*, Pasadena, CA, USA, 19–23 May 2008; pp. 455–460.

21. Hassani, A.; Morris, N.; Spenko, M. Experimental integrity evaluation of tightly-integrated IMU/LiDAR including return-light intensity data. In *Proceedings of the 32nd International Technical Meeting of The Satellite Division of the Institute of Navigation (ION)*, Miami, FL, USA, 16–20 September 2019; pp. 2637–2658.

22. Velas, M.; Spanel, M.; Hradis, M. CNN for IMU assisted odometry estimation using velodyne LiDAR. In *Proceedings of the 2018 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC)*, Torres Vedras, Portugal, 25–27 April 2018; pp. 71–77.

23. Deilamsalehy, H.; Havens, T.C. Sensor fused three-dimensional localization using IMU, camera and LiDAR. In *Proceedings of the 2016 IEEE Sensors*, Orlando, FL, USA, 30 October–3 November 2016; pp. 1–3.

24. Xie, G.; Zong, Q.; Zhang, X. Loosely-coupled lidar-inertial odometry and mapping in real time. *Int. J. Intell. Robot. Appl.* 2021, 5, 119–129. [CrossRef]
25. Moore, T.; Stouch, D. A generalized extended kalman filter implementation for the robot operating system. In *Intelligent Autonomous Systems 13*; Springer: Cham, Switzerland, 2016; pp. 335–348.
26. Ye, H.; Chen, Y.; Liu, M. Tightly coupled 3d lidar inertial odometry and mapping. In Proceedings of the 2019 IEEE International Conference on Robotics and Automation (ICRA), Montreal, QC, Canada, 20–24 May 2019; pp. 3144–3150.
27. Shan, T.; Englot, B.; Meyers, D. Lio-sam: Tightly-coupled lidar inertial odometry via smoothing and mapping. In Proceedings of the 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Las Vegas, NV, USA, 25–29 October 2020; pp. 5135–5142.
28. Wellington, C.; Stentz, A. Learning predictions of the load-bearing surface for autonomous rough-terrain navigation in vegetation. In *Springer Tracts in Advanced Robotics (STAR)*; Springer: Berlin/Heidelberg, Germany, 2003; Volume 24, pp. 83–92.
29. Bosse, M.; Zlot, R. Continuous 3D scan-matching with a spinning 2D laser. In Proceedings of the 2009 IEEE International Conference on Robotics and Automation (ICRA), Kobe, Japan, 12–17 May 2009; pp. 4312–4319.
30. Bosse, M.; Zlot, R.; Flick, P. Zebedee: Design of a spring-mounted 3-d range sensor with application to mobile mapping. *IEEE Trans. Robot.* **2012**, *28*, 1104–1119. [CrossRef]
31. Palieri, M.; Morrell, B. Locus: A multi-sensor lidar-centric solution for high-precision odometry and 3d mapping in real-time. *IEEE Robot. Autom. Lett.* **2020**, *6*, 421–428. [CrossRef]
32. Jiao, J.; Ye, H.; Zhu, Y.; Liu, M. Robust odometry and mapping for multi-lidar systems with online extrinsic calibration. *IEEE Trans. Robot.* **2021**, *1–10*. [CrossRef]
33. Nguyen, T.M.; Yuan, S. MILIOM: Tightly Coupled Multi-Input Lidar-Inertia Odometry and Mapping. *IEEE Robot. Autom. Lett.* **2021**, *6*, 5573–5580. [CrossRef]
34. Forster, C.; Carlone, L.; Dellaert, F. On-Manifold Preintegration for Real-Time Visual-Inertial Odometry. *IEEE Trans. Robot. Autom.* **2017**, *33*, 1–21. [CrossRef]
35. Qin, T.; Li, P.; Shen, S. Vins-mono: A robust and versatile monocular visual-inertial state estimator. *IEEE Trans. Robot.* **2018**, *34*, 1004–1020. [CrossRef]
36. Ceres Solver. Available online: http://ceres-solver.org (accessed on 5 October 2021).