

# Prior Map Aided LiDAR-Based Localization Framework via Factor Graph

Linqiu Gui<sup>1</sup>, Chunnian Zeng<sup>2</sup>, Jie Luo<sup>2,\*</sup> and Haotian Feng<sup>2</sup>

<sup>1</sup>*School of Information Engineering, Wuhan University of Technology, Wuhan, Hubei, PRC.*

<sup>2</sup>*School of Automation, Wuhan University of Technology, Wuhan, Hubei, PRC.*

## Abstract

Localization plays a vital role in unmanned platforms. The feature-based LiDAR odometry has excellent real-time performance but suffers accumulated error. The localization based on global matching can provide an unbiased estimation of states, whereas it requires an amount of calculation. This paper proposed a joint framework that fuses the LiDAR, IMU, and pre-build prior map to estimate the unmanned platform's six-degrees-of-freedom (DOF) states to solve the accumulated error problem under the premise of ensuring real-time. This work divided the system into two main threads. The one is utilizing the high-frequency feature-based LiDAR odometry coupled with IMU to localize. And the other is using the low-frequency global matching to correct the odometry. We link the two threads through sub-map optimization built atop the factor graph. As the global matching can provide unbiased estimation, and feature-based LiDAR odometry is with little calculation, the system can achieve both accuracy and real-time performance. The proposed framework is extensively validated with experiments under two platforms and achieves excellent performance.

## Keywords

Localization, prior map, global match, factor graph

## 1. Introduction

Localization is essential for most unmanned platforms such as automatic vehicles, moving robots, and unmanned air vehicles. In tradition, the localization usually relies on the collaborative cooperation of GPS and INS[1]. However, GPS signal is easy to lose due to the shelter like trees, rocks, and buildings; meanwhile, resulting from the accumulated error, the accuracy of the IMU is extremely low. Therefore, equipped with a perceptual sensor like LiDAR or camera on board, the odometry can be designed for localization and independent of external signals. Among them, the LiDAR-based methods are more robust and invariable to interference, and it becomes more convenient to be applied in practical applications.

There are usually two ways to implement odometry. One is SLAM, and the other is localization in a pre-built prior map through global matching.

LiDAR SLAM plays a pivotal role in unmanned platforms, which have witnessed rapid progress in the past few years[2]. Recently, fusing multiple sensors like IMU and GPS, the


---


*IPIN 2022 WiP Proceedings, September 5 - 7, 2022, Beijing, China*

\*Corresponding author.

✉ linqiugui1992@qq.com (L. Gui); ZengChn@whut.edu.cn (C. Zeng); luo\_jie@whut.edu.cn (J. Luo); fenghaotian@whut.edu.cn (H. Feng)

ORCID 0000-0003-1015-5834 (L. Gui); 0000-0002-6088-4982 (J. Luo)

 © 2022 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

 CEUR Workshop Proceedings (CEUR-WS.org)

LiDAR SLAM can provide accurate real-time state estimation and map, especially the tightly coupled method with IMU. However, the online localization of SLAM also suffers from drift when the GPS signal is lost consecutively. As a result, the precision is not enough to meet practical application requirements.

The global match is to match the point cloud of the current LiDAR frame against the prior map to estimate the current pose. The prior map is built previously by fusing the LiDAR odometry, GPS, and loop closure. Although the GPS signal is not fully covered, the system can establish a high precision prior map thanks to the globally consistent optimization. Usually, the global matching can be expressed as a nonlinear optimization problem, and it has specific requirements for a large number of calculations. Therefore it decreases the real-time performance and the frequency of localization. In addition, an excellent initial pose is also requested for global matching. However, Most of the existing methods utilize the linear model to estimate the initial pose, which will result in low precision and low robustness. The environmental change due to the dynamic object or seasonal variation will also affect the performance of global matching[3].

Aiming at the above problems, we propose a joint framework that fuses the LiDAR odometry with global matching to estimate the six-degrees-of-freedom (DOF) state for unmanned platforms. The LiDAR odometry obtained from feature-based scan matching is fast but with cumulative error. The global localization obtained from global matching is slow, but it can offer unbiased estimation with high precision. So the proposed framework utilizes the high-frequency LiDAR odometry coupled with IMU to localize and the low-frequency global matching to correct it so that the system can achieve both accuracy and real-time performance. We link the LiDAR odometry and the global matching through sub-map optimization atop the factor graph. The LiDAR odometry can also provide a tremendous initial pose for global matching. In addition, the LiDAR odometry matching against the current sub-map can effectively decrease the error result from the global matching in the changing environment. Because the prior map needs to be established in advance, my project is most suitable for specific scenes such as underground parking, scenic spots, and factory areas. Thanks to the global localization from the prior map, the system can achieve high-precision positioning without relying on GPS information so that the system can be applied to indoor localization tasks.

The main contributions of our work can be summarized as follows:

- We proposed a joint framework that utilizes the high-frequency LiDAR odometry coupled with IMU to localize and the low-frequency global matching to correct it. And, We link the LiDAR odometry and the global matching through sub-map optimization atop the factor graph. In this way, the system can achieve both accuracy and real-time performance.
- The proposed framework is extensively validated with experiments in two environments and achieves excellent performance in precision and robustness.

## 2. Related Work

Scan matching is the most important part of lidar odometry, and many algorithms deal with it; Iterative closest point (ICP)[4], and GICP[5] are the famous algorithms among them. However, matching a full point cloud is less computational efficient, so the feature-based matching methods are more suitable. The well-known method is LOAM proposed in [6], which extracts

edge and planar features by evaluating the roughness of points over a local region for scan-scan matching. Based on LOAM, Lego-loam proposed in [7] is ground-optimized and matches the plane feature and angle separately. Our work also follows the feature-based matching methods to ensure real-time performance.

Recently, LiDAR odometry is typically coupled with IMU for de-skewing the point cloud and joint optimization[8]. The coupling method can mainly be grouped into loosely coupled and tightly coupled. The popular approach for loosely-coupled is using extended Kalman filters (EKF)[9, 10, 11, 12]. However, the tightly-coupled method especially based on the IMU preintegration algorithm proposed in [13] usually offers improved performance and has attracted wide attention[14]. For example, In [15], a tightly coupled lidar- inertial odometry algorithm is proposed and provides real-time accurate state estimation with a high update rate. In [8], the LIO framework is built atop a factor graph and is suitable for multi-sensor fusion and global optimization, achieving excellent performance. Our proposed system also followed this work.

Global matching is essential for localization. In [16], the intensity and altitude are adaptively combined to match the online lidar scan against the prior map and conduct a 3 DOF (x,y, yaw) estimation. As to 6 DOF estimation, In [17], the ICP is utilized for global matching. However, it is known that the ICP is very computationally inefficient and sensitive to environmental change. The Normal Distribution Transform (NDT) method proposed in [18] has been more popular in recent years. It has both advantages of real-time and precision. In addition, it has more tolerance for the accuracy of the initial pose. So, our work also follows the NDT method for global matching.

### 3. System Overview

#### 3.1. Definition and Notation

The mathematical symbols used in this paper are defined in Table 1.

**Table 1**  
Mathematical Symbols

Symbols	Name
$T$	Transpose
$N$	Normal distribution
$SO(3)$	Special Orthogonal Group
$SE(3)$	Special Euclidean Group
$\cup$	Intersection
$\in$	Belong To
$\mathbb{R}$	Real Number
$\doteq$	Or
$\sum$	Summation
$\prod$	Product Operator
$\ *\ $	Mahalanobis Distance

We define the skew symmetric matrix of a vector  $\omega \in \mathbb{R}^3$  using  $(\cdot)^\wedge$ :

$$\omega^\wedge = \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix}^\wedge = \begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix}. \quad (1)$$

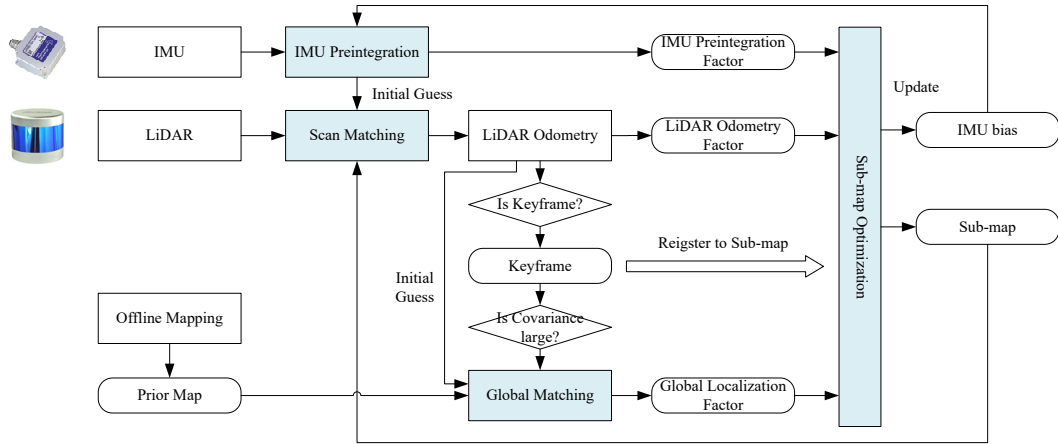
Similarly, we can map a skew symmetric matrix to a vector in  $\mathbb{R}^3$  using the operator  $(\cdot)^\vee$ . We use  $R \in SO(3)$ ,  $p \in \mathbb{R}^3$ , and  $T = [R|p] \in SE(3)$  to represent the rotation matrix, the position vector, and the transformation matrix, respectively. It is worth mentioning that the transformation matrix from the reference coordinate to the object coordinate can also describe the object's pose in the reference coordinate. We denote the world frame by  $\mathbf{W}$ , and it is the same as East, North, and Up (ENU) coordinates. We denote the platform body frame by  $\mathbf{B}$  that is the same as the IMU frame. We also define the GPS frame by  $\mathbf{G}$  and the LiDAR frame by  $\mathbf{L}$ . Assume that the parameters in frame  $\mathbf{A}$  can be wrote as  $(\cdot)^A$  and the transformation from frame  $\mathbf{B}$  to frame  $\mathbf{A}$  can be denoted as  $(\cdot)^{AB}$ . The state of the platform  $X$  can be defined as:

$$X = [R^\top, p^\top, v^\top, b^\top]^\top \quad (2)$$

where  $v \in \mathbb{R}^3$  denotes the speed,  $b = [b^a, b^\omega]$  denotes the IMU bias,  $b^a \in \mathbb{R}^3$  is the acceleration bias, and  $b^\omega \in \mathbb{R}^3$  is the gyroscope bias.

### 3.2. System Structure

This paper proposed a framework that fuses the LiDAR odometry, the IMU preintegration, and the global localization to estimate the six degrees-of-freedom(DOF) pose. The architecture of the proposed framework is shown in Figure.1, and it mainly consists of four modules: Scan Matching, IMU preintegration, Global Matching, and Sub-map Optimization. We marked them blue in the figure.



**Figure 1:** The system pipeline

The Scan Matching module provides the LiDAR odometry by matching the current LiDAR scan to the sub-map. In our system, we follow the feature-based scan matching method proposed

in [6] for computational efficiency. Using every LiDAR frame for optimization is computationally intractable. Instead, we consider the keyframes and use a simple but widely used approach for keyframe selection, which registers a LiDAR frame as a keyframe when its pose exceeds a user-defined threshold compared with the previous keyframe. The data of the  $n$ th keyframe from LiDAR can be written as  $\mathbb{F}_n$ , and the related state is defined as  $X_n$ . The IMU preintegration module is built by following Christian Forster's work proposed in [13], and it mainly provides the IMU preintegration factor for sub-map optimization and provides the initial guess for Scan Matching at each computation cycle. The Global Matching module provides the global localization factor for the sub-map optimization. The Global Matching module is built by following the NDT (Normal-Distributions Transform) method proposed in [19] by Magnusson, and it can conduct an unbiased estimation of the pose through matching online LiDAR scans against the prior map built in advance. The sub-map contains the feature points of a fixed number of resending LiDAR keyframes. The optimization of the sub-map can directly affect the accuracy of Lidar Odometry. So we fuse the LiDAR odometry factor, IMU preintegration factor, and Global Matching Factor for optimization through a factor graph. The optimization result will update the IMU bias and the LiDAR odometry.

#### 4. LiDAR odometry

We follow the scan matching method proposed in [6]. It utilizes the feature points rather than ICP to achieve scan matching. Therefore it greatly improves the real-time performance. In addition, we follow the scan to the sub-map matching frame proposed in [8], it can make the most of sub-keyframe constraints to improve the precision.

When the LiDAR frame  $\mathbb{F}_n$  is received, we first calculate the roughness value of each point by comparing it to the consecutive neighbor points in the same scan. Then the edge and planar features will be extracted according to the roughness value. The points with large values are classified as the edge features and denoted as  $F_n^e$  and the points with small roughness are classified as the planar feature and denoted as  $F_n^p$ . The features of the LiDAR frame  $\mathbb{F}_n$  are in frame  $\mathbf{L}$ . Then the extracted features  $\{F_n^e, F_n^p\}$  are matched to the features from sub-map as  $M_{n-1} = \{M_{n-1}^e, M_{n-1}^p\}$ .  $M_{n-1}$  gathers the features of the previous keyframes, and the features are transformed to  $\mathbf{W}$  by LiDAR pose:

$$\begin{aligned} M_{n-1}^e &= \bar{F}_{n-1}^e \cup \bar{F}_{n-2}^e \cup \dots \cup \bar{F}_{n-N_k-1}^e \\ M_{n-1}^p &= \bar{F}_{n-1}^p \cup \bar{F}_{n-2}^p \cup \dots \cup \bar{F}_{n-N_k-1}^p \end{aligned} \quad (3)$$

where  $\bar{F}_n^e$  and  $\bar{F}_n^p$  are the transformed edge and planar features in  $\mathbf{W}$ . The distance of features between the current scan and sub-map are  $\{d_e(\tilde{T}_n p_{n,i}^e), d_p(\tilde{T}_n p_{n,j}^p)\}$ ,  $p_{n,i}^e \in F_n^e, p_{n,j}^p \in F_n^p$ ,  $\tilde{T}_n$  is the transformation of keyframe  $\mathbb{F}_n$  to be estimated, and the detail definition of distance can be seen in [6]. As the result, the optimization problem can be written as:

$$\hat{T}_n^{WL} = \arg \min_{\tilde{T}_n} \left\{ \sum_{p_{n,i}^e \in F_n^e} d_e(\tilde{T}_n p_{n,i}^e) + \sum_{p_{n,j}^p \in F_n^p} d_p(\tilde{T}_n p_{n,j}^p) \right\} \quad (4)$$

The initial guess of  $\tilde{T}_n$  can be obtain from the IMU preintegration in Section 5. Through manifold optimization[20] and GaussNewton method, we can solve it to obtain the transformation  $\hat{T}_n^{WL}$ . Given the extrinsic parameters between  $\mathbf{L}$  and  $\mathbf{B}$  as  $T^{BL}$ , then we can calculate the pose of platform  $\hat{T}$  and the relative transformation  $\Delta\hat{T}_{n-1,n}$  between the  $\mathbb{F}_{n-1}$  and  $\mathbb{F}_n$  through the equation:

$$\begin{aligned}\hat{T}_n &= \hat{T}_n^{WL} (T^{BL})^\top \\ \Delta\hat{T}_{n-1,n} &= (\hat{T}_{n-1})^\top \hat{T}_n\end{aligned}\quad (5)$$

The  $\Delta\hat{T}_{n-1,n}$  can be seen as the measurement from LiDAR, assumed that it's noise is  $\delta\xi$ , it can be written as:

$$\Delta\hat{T}_{n-1,n} = \Delta T_{n-1,n} \exp(\delta\xi_{n-1,n}^\wedge) \quad (6)$$

The residual errors of scan matching can be defined as:

$$r_{LiDAR} = [\delta\xi_{n-1,n}] = \log((\Delta T_{n-1,n})^\top \Delta\hat{T}_{n-1,n})^\vee \quad (7)$$

## 5. IMU preintegration

To avoid repeatedly integrating caused by the changes in initial conditions when the lidar odometry pose has been optimized, we adopt preintegration of the inertial measurements introduced by [13] in our implementation.

Firstly, taking the accelerator bias  $b^a$ , gyroscope bias  $b^\omega$  and the additive noise into consider, the IMU measurements formula can be written as follow:

$$\begin{aligned}\hat{a}_t &= R_t^\top (a_t - g) + b_t^a + n_t^a \\ \hat{\omega}_t &= \omega_t + b_t^\omega + n_t^\omega.\end{aligned}\quad (8)$$

Where  $t$  deonte one sampling time. Where the  $g$  denotes the gravity vector in W. The  $n_t^a$  and  $n_t^\omega$  denotes the additive noise in acceleration and gyroscope measurements, them can be seen as the Gaussian white noise as  $n_t^a \sim N(0, \sigma_a^2)$ ,  $n_t^\omega \sim N(0, \sigma_\omega^2)$ . Combining the integration of the IMU measurements with the motion formula of platform, and we discretize and iterate it for intervals between sampling times  $i$  and  $j$ , we have:

$$\begin{aligned}R_j &= R_i \prod_{k=i}^{j-1} \exp((\hat{\omega}_k - b_k^\omega - n_k^\omega)^\wedge \Delta t) \\ v_j &= v_i + g\Delta t_{ij} + \sum_{k=i}^{j-1} R_k (\hat{a}_k - b_k^a - n_k^a) \Delta t \\ p_j &= p_i + \sum_{k=i}^{j-1} \left[ v_k \Delta t + \frac{1}{2} g \Delta t^2 + \frac{1}{2} R_k (\hat{a}_k - b_k^a - n_k^a) \Delta t^2 \right].\end{aligned}\quad (9)$$

Where the  $\Delta t$  is the interval between two consecutive IMU measurements. We can defined the preintegrated items as:

$$\begin{aligned}\Delta R_{ij} &\doteq R_i^\top R_j \\ \Delta v_{ij} &\doteq R_i^\top (v_j - v_i - g\Delta t_{ij}) \\ \Delta p_{ij} &\doteq R_i^\top (p_j - p_i - v_i \Delta t_{ij} - \frac{1}{2}g\Delta t_{ij}^2)\end{aligned}\tag{10}$$

Assuming that the bias remains constant between sampling times  $i$  and  $j$ , the preintegrated measurements  $\Delta \hat{p}_{ij}$ ,  $\Delta \hat{v}_{ij}$ ,  $\Delta \hat{R}_{ij}$  can be obtained as:

$$\begin{aligned}\Delta \hat{R}_{ij} &\doteq \prod_{k=i}^{j-1} \exp((\hat{\omega}_k - b_i^\omega)^\wedge \Delta t) = \Delta R_{ij} \exp((\delta \phi_{ij})^\wedge) \\ \Delta \hat{v}_{ij} &\doteq \sum_{k=i}^{j-1} \Delta \hat{R}_{ik} (\hat{a}_k - b_i^a) \Delta t = \Delta v_{ij} + \delta v_{ij} \\ \Delta \hat{p}_{ij} &\doteq \sum_{k=i}^{j-1} \left[ \Delta \hat{v}_{ik} \Delta t + \frac{1}{2} \Delta \hat{R}_{ik} \cdot (\hat{a}_k - b_i^a) \Delta t^2 \right] \\ &= \Delta p_{ij} + \delta p_{ij}.\end{aligned}\tag{11}$$

where the  $\delta \phi_{ij}$ ,  $\delta v_{ij}$ ,  $\delta p_{ij}$  denote the noise of  $\Delta \hat{R}_{ij}$ ,  $\Delta \hat{v}_{ij}$ ,  $\Delta \hat{p}_{ij}$  respectively and the detail definition can be seen in [13]. Considering the update of IMU bias, we can obtain the residual errors between two LiDAR scan ( $X_{n-1}$ ,  $X_n$ ) as:

$$r_{imu} = \begin{bmatrix} \delta \phi_{n-1,n} \\ \delta v_{n-1,n} \\ \delta p_{n-1,n} \\ \delta b_{n-1,n}^a \\ \delta b_{n-1,n}^\omega \end{bmatrix} = \begin{bmatrix} \log(\Delta(R_{n-1,n})^\top \Delta \hat{R}_{n-1,n})^\vee \\ \Delta \hat{v}_{n-1,n} - \Delta v_{n-1,n} \\ \Delta \hat{p}_{n-1,n} - \Delta p_{n-1,n} \\ b_n^a - b_{n-1}^a \\ b_n^\omega - b_{n-1}^\omega \end{bmatrix}.\tag{12}$$

## 6. Global Matching

The global matching procedure mainly provides the no drift pose estimation for sup-map modules through matching online lidar scan against the prior map, which is built offline.

Offline mapping plays an essential role in our system because it provides the absolute correction data to eliminate the drift. As a result, the quality of the prior map will directly influence the precision of the localization in online procedures. The offline mapping procedure follows the SLAM method proposed in [8] which fuses the LiDAR, IMU, and GPS for establishing the point cloud map. Thanks to the globally consistent optimization, we can get a high precision map with limited GPS and loop closure.

We utilize the NDT (Normal-Distributions Transform) method proposed by Martin Magnusson[18] for global matching, which has more real-time performance than the iterative closest point (ICP) method.

Firstly, we grid the prior map into a mass of 3x3 small cells. For each cell, it is assumed that there are  $j$  points  $p_{k=1,\dots,j} \in \mathbb{R}^3$  in it and the points conform to the normal distribution relationship. Therefore, the normal distribution parameter can be calculated based on the points in the cell, and we can easily get the probability density function (PDF) of it:

$$\begin{aligned}\mu &= \frac{1}{j} \sum_{k=1}^j p_k \\ \Sigma &= \frac{1}{j} \sum_{k=1}^j (p_k - \mu)(p_k - \mu)^T\end{aligned}\tag{13}$$

Where the  $\mu$  and  $\Sigma$  denote the mean and the covariance of the normal distribution, respectively. The probability density function of the cell can be written as:

$$f_{ndt}(x) = \frac{1}{(2\pi)^{\frac{3}{2}} \sqrt{|\Sigma|}} e^{-\frac{(x-\mu)^T \Sigma^{-1} (x-\mu)}{2}}\tag{14}$$

When a lidar scan  $\mathbb{F}_n = \{P_{k=1,\dots,i}^L\}$  is about to match against the prior map, the goal is to find out its translation of lidar  $T_n^{WL}$  to maximize the likelihood that all of the lidar scan points will be on the cell of reference points cloud map. Therefore, combining the probability density function of the cell in (14), the likelihood function can be easily written as:

$$Likelihood : \theta = \prod_{k=1}^i f_{ndt}^k(T_n^{WL} P_k^L)\tag{15}$$

It's worth noting that the  $f_{ndt}^k$  is the probability density function of the cell which the  $k$ th point  $T_n^{WL} P_k^L$  located. This's a nonlinear optimization problem, and we can use the Newton optimization to solve it and obtain the optimized translation of LiDAR. The initial guess can be obtained from the LiDAR odometry in Section 4. Through the transformation of the coordinate system  $T_n = T_n^{WL} (T^{BL})^\top$ , we can obtain the pose estimation of platform as  $\hat{T}_n$ . Assumed that the noise of  $T_n$  is  $\delta\xi^{ndt}$ , we have:

$$\hat{T}_n = T_n \exp((\delta\xi_n^{ndt})^\wedge)\tag{16}$$

The residual errors of scan matching can be defined as:

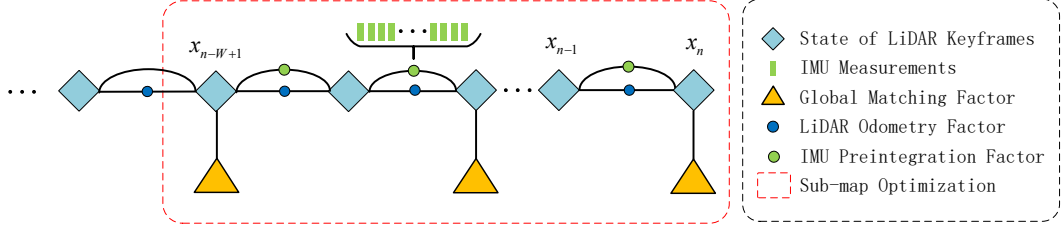
$$r_{ndt} = [\delta\xi_n^{ndt}] = \log((T_n)^\top \hat{T}_n)^\vee\tag{17}$$

In practical application, the NDT method is already integrated into the PCL library[21] and can be used conveniently.

## 7. Sub-map Optimization

According to the Section 4, the sub-map  $M_n$  can be defined as the collection of the features of previous keyframes, and the features are transformed to  $\mathbf{W}$  by LiDAR pose. Therefore, we only





**Figure 2:** The illustration of system optimization procedure

need to optimize the states of all the keyframes in the sub-map. The state need to be optimized can be defined as:

$$\chi = \{X_m, X_{m+1}, \dots, X_n\} \quad (18)$$

we utilize the factor graph to model this problem. Fig. 2 provides a brief illustration of optimization procedure. The keyframe state can be seen as the node, and we fusing the LiDAR odometry factor, IMU preintegration factor and the global matching factor for optimizing. The residual error function can be written as:

$$\min_{\chi} \left\{ \sum_{k \in \psi} \|r_{imu}(z_k^I, \chi)\|^2 + \sum_{k \in \tau} \|r_{LiDAR}(z_k^L, \chi)\|^2 + \sum_{k \in \vartheta} \|r_{ndt}(\chi)\|^2 \right\}. \quad (19)$$

The  $r_{LiDAR}$ ,  $r_{imu}$  and  $r_{ndt}$  are residuals for LiDAR odometry, IMU preintegration and global matching, which are defined in Section 4, Section 5 and Section 6 respectively.

In addition, to decrease the calculation, we utilize an unfixed sliding window to register the keyframes for the sub-map, and we also use a simple but efficient way for marginalization. We register the keyframe into the window when it is received. If the covariance of the latest keyframe is larger than a set threshold, we perform a global match on it. Then, the keyframes before the second oldest NDT factor will be marginalized out of the window. Because the NDT has already provided the prior value, we don't need to convert measurements corresponding to marginalized states into a prior. When the global matching of the latest keyframe is done, the optimization process is performed. After that, the LiDAR odometry and the IMU bias will be updated according to the optimization results.

Due to the long time consuming of global matching and optimization, the optimization process will lag. However, the global matching, the optimization process, and the LiDAR odometry belong to different threads, so the optimization lag does not affect the operation of LiDAR odometry. Since LiDAR Odometry is processed by matching the scan to the sub-map, the accuracy of Lidar Odometry can be improved by optimizing the sub-map. Meanwhile, thanks to the existence of a global matching factor in sub-map optimization, LiDAR odometry can achieve unbiased estimation.

The sub-map optimization can be represented in pseudocode as Algorithm. 1.

---

**Algorithm 1** Sub-map optimization procedure

---

```
1: while 1 do
2:   if A now LiDAR keyframe received then
3:     Register LiDAR keyframe into sliding window.
4:     if Covariance larger than setting threshold? then
5:       NDT based Global matching (See Sec.6).
6:       Marginalization.
7:       Add NDT factor.
8:       Sub-map optimization (See Sec.7).
9:       Update the LiDAR odometry and the IMU bias.
10:    end if
11:  end if
12: end while
```

---

## 8. Experiments

### 8.1. Platforms and Datasets

We collected several data sets on the WHUT campus to analyze the proposed framework qualitatively and quantitatively. The sensor suite used in our work includes an ouster OS1-32 and an RTK GPS module, the OS1-32 has a built-in IMU, and the internal parameters are calibrated by the method proposed in [22], the RTK GPS provides the ground-truth for quantitative analysis. For validation, we collected six different datasets across two platforms: A-M, A-T-1, A-T-2, B-M, B-T-1, and B-T-2. The A and B mean the data sequences A and B, respectively, the M means the mapping datasets, and the T means the test datasets. The platform of A data sequence is the low-speed Logistics vehicle, the lidar is installed atop it, and the RTK GPS is installed at the same horizontal position. The platform of the B data sequence is the high-speed electric vehicle, and the lidar is installed ahead of it, so we only get half of the points and meet the extreme condition. The platforms are illustrated in Fig.3.

Our method and comparisons are implemented in C++ and executed on a JETSON TX2 with the framework of Robot Operating System (ROS)[23] in Ubuntu Linux.

### 8.2. Offline Mapping

Firstly, we utilized the A-M and B-M for mapping, the result is shown in table.2, and the trajectory with mapping error is shown in Fig.4.

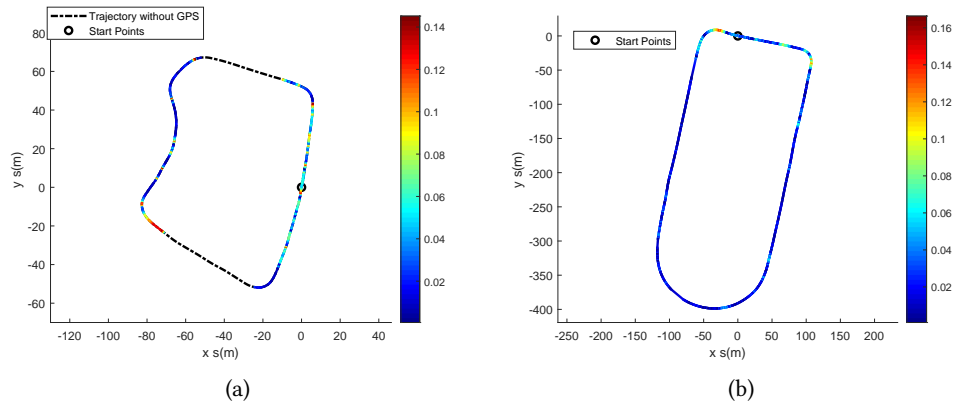
As shown in Fig.4(a), GPS data is missing on some roads in data sequence A, and those were taken out of the analysis. The average speed within a data sequence is the same. The data sequence B only has half the point cloud, and the moving speed is faster. As a result, its accuracy is relatively low. It can be seen that the accuracy of the mapping is approximately in centimeters and meet the requirements.



**Figure 3:** The experimental platforms illustration: (a) platform A; (b) platform B;

**Table 2**  
Mapping details

Dataset	Speed (Km/h)	Trajectory length (m)	RMSE (m)	Max Error (m)	Size (MB)
A-M	3.8	342	0.0566	0.1601	197.0
B-M	32.73	1075.6	0.1293	0.3582	233.9



**Figure 4:** The trajectory with mapping error: (a) A-M; (b) B-M;

### 8.3. Localization

Then, we used the datasets A-T-1, A-T-2, B-T-1, and B-T-2 to analyze the online location module on the prior map they built, respectively. Our method is comparable to the LIO method from [8] and the NDT method from [19] under the main metrics of a translation error. The results

are shown in Table.3, the unit for the values is (m). It is worth noting that the translation error only considers the x, y axis.

**Table 3**  
Experimental results

Data sequences	DataSets	Metrics	LIO	NDT	Our
A	A-T-1	RMSE	1.4271	0.082	<b>0.0749</b>
		MAX	2.9252	0.1871	<b>0.1605</b>
	A-T-2	RMSE	2.1336	<b>0.0947</b>	0.1046
		MAX	3.3551	0.2339	<b>0.2137</b>
B	B-T-1	RMSE	2.0689	Fail	<b>0.2090</b>
		MAX	3.8010	Fail	<b>0.4658</b>
	B-T-2	RMSE	5.5463	0.3109	<b>0.2059</b>
		MAX	9.5560	0.9306	<b>0.4495</b>

It can be seen that the LIO method has the most significant error cause it suffers from drift. The NDT method used the linear model to estimate the initial guess based on the assumption of the short-term velocity invariance. The imprecise initial guess results in a bigger error, especially in cases like a sharp turning point and Speed-up Speed-down, which will cause the failure like the dataset B-T-1. In contrast, our method achieves better precision in most datasets due to the fusion of LIO and NDT methods. However, our approach also relies on NDT to eliminate accumulated errors, so it has no significant improvement in accuracy than the NDT method but significantly improves robustness, especially in extreme cases.

## 9. Conclusions And Discussion

In this paper, we propose a precision and robust LiDAR localization framework that combines LiDAR odometry with NDT-based global matching. We link them through a sub-map optimization built atop the factor graph. Through the experimental analysis, it has been shown that our method can perform well. It's worth noting that the problem discussed in this paper does not involve estimating the initial position on the map, which is usually given by GPS or related algorithms in engineering. Since GPS is not needed for the proposed localization system, so the system is also suitable for indoor environments. However, due to the limitation of experimental conditions, the experiments in this paper are carried out outdoors. Follow-up indoor experiments are required. Although we have achieved some phased results, there is still a lot of work to do, such as the influence of environmental changes on localization and the dynamic update of the map. We will handle those problems in future work.

## References

- [1] H. Qi, J. Moore, Direct kalman filtering approach for gps/ins integration, IEEE Transactions on Aerospace and Electronic Systems 38 (2002) 687–693.

- [2] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, J. J. Leonard, Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age, *IEEE Transactions on robotics* 32 (2016) 1309–1332.
- [3] W. Ding, S. Hou, H. Gao, G. Wan, S. Song, Lidar inertial odometry aided robust lidar localization system in changing city scenes, in: *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 2020, pp. 4322–4328.
- [4] P. Besl, H. McKay, A method for registration of 3-d shapes, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 14 (1992) 239–256.
- [5] A. Segal, D. Haehnel, S. Thrun, Generalized-icp., in: *Robotics: science and systems*, volume 2, Seattle, WA, 2009, p. 435.
- [6] J. Zhang, S. Singh, Low-drift and real-time lidar odometry and mapping, *Autonomous Robots* 41 (2017) 401–416.
- [7] T. Shan, B. Englot, 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, 2018, pp. 4758–4765.
- [8] T. Shan, B. Englot, D. Meyers, W. Wang, C. Ratti, R. Daniela, Lio-sam: Tightly-coupled lidar inertial odometry via smoothing and mapping, in: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2020, pp. 5135–5142.
- [9] S. Lynen, M. W. Achtelik, S. Weiss, M. Chli, R. Siegwart, A robust and modular multi-sensor fusion approach applied to mav navigation, in: *2013 IEEE/RSJ international conference on intelligent robots and systems*, IEEE, 2013, pp. 3923–3929.
- [10] S. Yang, X. Zhu, X. Nian, L. Feng, X. Qu, T. Ma, A robust pose graph approach for city scale lidar mapping, in: *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2018, pp. 1175–1182.
- [11] M. Demir, K. Fujimura, Robust localization with low-mounted multiple lidars in urban environments, in: *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, IEEE, 2019, pp. 3288–3293.
- [12] Y. Gao, S. Liu, M. M. Atia, A. Noureldin, Ins/gps/lidar integrated navigation system for urban and indoor environments using hybrid scan matching algorithm, *Sensors* 15 (2015) 23286–23302.
- [13] C. Forster, L. Carlone, F. Dellaert, D. Scaramuzza, On-manifold preintegration for real-time visual-inertial odometry, *IEEE Transactions on Robotics* 33 (2016) 1–21.
- [14] C. Chen, H. Zhu, M. Li, S. You, A review of visual-inertial simultaneous localization and mapping from filtering-based and optimization-based perspectives, *Robotics* 7 (2018) 45.
- [15] H. Ye, Y. Chen, M. Liu, Tightly coupled 3d lidar inertial odometry and mapping, in: *2019 International Conference on Robotics and Automation (ICRA)*, 2019, pp. 3144–3150.
- [16] G. Wan, X. Yang, R. Cai, H. Li, Y. Zhou, H. Wang, S. Song, Robust and precise vehicle localization based on multi-sensor fusion in diverse city scenes, in: *2018 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2018, pp. 4670–4677.
- [17] K. Yoneda, H. Tehrani, T. Ogawa, N. Hukuyama, S. Mita, Lidar scan feature for localization with highly precise 3-d map, in: *2014 IEEE Intelligent Vehicles Symposium Proceedings*, IEEE, 2014, pp. 1345–1350.
- [18] M. Magnusson, The three-dimensional normal-distributions transform: an efficient representation for registration, surface analysis, and loop detection, Ph.D. thesis, Örebro

universitet, 2009.

- [19] The three-dimensional normal-distributions transform : an efficient representation for registration, surface analysis, and loop detection, renewable energy (2009).
- [20] F. Dellaert, M. Kaess, et al., Factor graphs for robot perception, *Foundations and Trends® in Robotics* 6 (2017) 1–139.
- [21] R. B. Rusu, S. Cousins, 3d is here: Point cloud library (pcl), in: 2011 IEEE international conference on robotics and automation, IEEE, 2011, pp. 1–4.
- [22] W. Gao, imu\_utils, [https://github.com/gaowenliang/imu\\_utils](https://github.com/gaowenliang/imu_utils), 2018.
- [23] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, A. Y. Ng, Ros: an open-source robot operating system, in: ICRA workshop on open source software, volume 3, Kobe, Japan, 2009, p. 5.