

# LIDAR-INERTIAL LOCALIZATION WITH GROUND CONSTRAINT IN A POINT CLOUD MAP

Mengchi Ai<sup>1</sup>, Ilyar Asl Sabbaghian Hokmabadi<sup>1</sup>, Mohamed Elhabiby<sup>2</sup>, Mohamed Moussa<sup>2</sup>, Ahmed Zekry<sup>2</sup>, Ahmed Mohamed<sup>1</sup>, Naser El-Sheimy<sup>1\*</sup>

<sup>1</sup>Dept. of Geomatics Engineering, University of Calgary, 2500 University Dr NW, Calgary, AB T2N 1N4, Canada – ([mengchi.ai](mailto:mengchi.ai), [ilyar.aslsabbaghianh](mailto:ilyar.aslsabbaghianh), [ahmed.mohamed2](mailto:ahmed.mohamed2), [@ucalgary.ca](mailto:elsheimy))

<sup>2</sup>Micro Engineering Tech. Inc., Calgary, AB, Canada, ([elhabiby](mailto:elhabiby), [m.moussa](mailto:m.moussa), [@microengineering.ca](mailto:ahmed.ayman))

**KEY WORDS:** Map-aided localization, Factor graph optimization, Point cloud map, Ground constraints, Sensor fusion

## ABSTRACT:

Real-time localization is a crucial task in various applications, such as automatic vehicles (AV), robotics, and smart city. This study proposes a framework for map-aided LiDAR-inertial localization, with the objective of accurately estimating the trajectory in a point clouds map. The proposed framework addresses the localization problem through a factor graph optimization (FGO), enabling the fusion of homogenous measurements for sensor fusion and designed absolute and relative constraints. Specifically, the framework estimates the light detection and ranging (LiDAR) odometry by leveraging inertial measurement unit (IMU) and registering corresponding featured points. To eliminate the accumulative error, this paper employs a ground plane distance and a map matching error to constraint the positioning error along the trajectory. Finally, local odometry and constraints are integrated using a FGO, including LiDAR odometry, IMU pre-integration, and ground constraints, map matching constraints, and loop closure. Experimental results were evaluated on an open-source dataset, UrbanNav, with an overall localization accuracy of 2.29 m (root mean square error, RMSE).

## 1. INTRODUCTION

Localization is a fundamental task for AVs as it provides real-time position to higher-level functions and automatics, such as vehicle control, path planning, and environment perceiving. The solution of LiDAR-based navigation coupled with inertial sensors, has become a widely used observation multi-sensors system for navigation tasks, as the observed point clouds are invariant to changing illumination and viewpoints (Ai et al., 2022). Although map-less LiDAR-based simultaneous localization and mapping (SLAM), such as Cartographer and LOAM-variants (Nüchter et al., n.d.; Ji Zhang and Singh, 2014), can generate accurate global mapping with real-time estimated trajectory, they are prone to misalignments and accumulated errors when there are limited stable points with associated features (Dube et al., 2017; Rozenberszki and Majdik, 2020; Schaefer et al., 2019). Therefore, involving a prior map which encodes the environmental features has become a popular solution to enhance the performance of long-term localization in urban canyon environments.

Researchers have successfully implemented map-aided navigation in urban environments; however, the performance is still limited in complicated scenarios. The challenging issues of map-aided LiDAR-based navigation can be discussed in two aspects. First, the accuracy of the odometry estimation is reduced when moving objects exist in the observed environments, as misalignments cannot be corrected by global constraints, such as loop closure, especially in cases where there are no loops along the trajectory. Although deep-learning-based semantic features have been widely used to enhance the alignment by involving object-level semantic patches, the positioning is still drifting overtime as the error continues to accumulate when the error source is not related to semantics. Secondly, the localization performance highly depends on the fusion scheme and designed constraints. The estimated trajectory is prone to falling into local minima resulting in unexpected errors. These challenging issues, if not addressed appropriately, can decrease the accuracy of LiDAR-based navigation, particularly in dense urban environments.

To address the aforementioned challenges, we propose a map-aided tightly coupled LiDAR-Inertial localization framework that incorporates ground constraints and map matching constraints to obtain a global trajectory. Specifically, relative odometry is estimated using LiDAR odometry, IMU pre-integration. Along the trajectory, this paper proposes to detect and apply the local absolute constraints including map matching constraint and ground constraint, as well as loop closure constraint to enhance the positioning. Finally, a FGO is designed to fuse all the constraints and estimated relative odometry and output the real-time high accuracy trajectory. The main contributions of this work are summarized as follows:

- We propose a map-aided navigation solution that integrates a LiDAR and an IMU for global localization and formulates the localization problem as FGO framework.
- To enhance the performance of localization, we introduce absolute constraints based on the segmented ground planes and the map matching, which are jointly optimized in the FGO.
- We present experimental evaluations that demonstrate the stable localization performance of our proposed framework in various urban environments.

## 2. RELATED WORK

In this section, we provide a brief review of map-aided LiDAR-based localization. We categorize related works into two aspects: the mathematical formulation of navigation and the algorithms of constraints.

Some researches propose place recognition algorithms that match online point clouds with the prior map for real-time positioning. For example, the Autoware implements NDT matching for LiDAR-based navigation and introduces a scan matching framework, that the performance highly depends on the map resolution (Kato et al., 2018). In (H. Liu et al., 2019), a multi-group-LM (Levenberg–Marquardt) optimization algorithm is

proposed to calculate the Jacobian matrix and avoid local minima. Other researchers propose error models such as Monte Carlo (MC) algorithm to fuse homogenous variables and establish state function. For example, in the case of (Dong et al., 2021), the localization framework is implemented based on MC localization, and online trajectory is estimated by matching extracted pole-like objects with the prior pole map and fusing the variables into particle filters. Similarly, in (Liu et al., 2022), an Extended Kalman Filter (EKF) and an adaptive MC algorithm are designed to fuse multi-source data to estimate relative motion. Inspired by FGO-based SLAM algorithms, we involve the FGO framework to estimate the trajectory, which is one of the most stable frameworks to fuse heterogeneous measurements and variables.

To address misalignment issues resulting in large drift along the trajectory, some researchers focus on feature encoding and extraction of the prior map to provide geometric description for the registration with onboarding sensors. For example, feature-based points, such as scale-invariant feature transform (SIFT) and infrared remittance, are extracted to represent the geometric distribution of surrounding environments (Imanullah et al., 2019; Levinson and Thrun, 2010). In (Bársan et al., 2020), the work introduces the intensity-based map and involves the joint embedded features for association. High-definition (HD) maps, such as OpenstreetMap, provide the distribution and positions of traffic signs, which are considered as geo-referenced objects and matched with perceived results from LiDAR (Cho et al., 2022). Other researchers involve training-based semantics such as DeepICP and Segmatch for the semantics-based alignment (Dube et al., 2017; Lu et al., 2019). However, the training models need to be updated with extra manual works when new environments are expected to be used as test datasets. In this work, we introduce a feature map that employs stable geometric points, including corner points, planar points and segmented ground points (Ji Zhang and Singh, 2014).

To enhance the robustness of long-term navigation, some researchers focus on designing constraints and optimization conditions for fault detection engines and boundary minimization. For example, overlapNet is proposed to estimate the yaw angle and recognize the loop closure for global mapping by aligning the semantics between LiDAR and the semantics map (Chen et al., 2022). Similarly, (Dube et al., 2017) introduces a semantics-aided matching algorithm for the loop closure, while (Zhu et al., 2020) introduces a graph-of-semantics network that involves semantic segments and generates graph descriptor for loop detection. In (X. Liu et al., 2019), ground points are segmented to provide consistent normal vector to eliminate the error from non-stationary objects. These works can address the problem by designing global constraints for accurate loop closure, however, the instability of point-level alignment is ignored during the local odometry estimation. Different from the previous works that focus on loop closure, this work proposes the local constraints

including the map matching and consistent ground plane to eliminate the drift during the movements.

### 3. METHODOLOGY

In this section, the detail of the proposed map-aided localization framework is introduced as follows.

#### 3.1 System overview

Considering a multi-sensor system comprising an onboard LiDAR and IMU, the system receives synchronized raw data from these sensors, and the measurement of Global Navigation Satellite System (GNSS) is optional. A prior feature map is geo-referenced and is composed of geometric feature points. In addition, it is assumed that all the sensors are pre-calibrated, and the IMU frame coincides with the body frame. To estimate the trajectory, the motion state  $x$  of the system can be formulated as follows:

$$x = [R^T, p^T, v^T, b^T]^T, \quad (1)$$

where  $R$  = rotation matrix,  $R \in SO(3)$   
 $p$  = position vector,  $p \in R^3$   
 $v$  = speed  
 $b$  = IMU bias

The objective is to estimate the transformation matrix  $T$  from body coordinate  $B$  to world coordinate  $W$ , which is represented as  $T_B^W = [R | p]$ .

Figure 1 provides an overview of the proposed localization framework, which aims to estimate the state node of the navigation in each timestamp. The state estimation is formulated as a maximum a posteriori (MAP) problem. In this study, a factor graph is used to model the MAP problem, which is one of the stable solution to fuse heterogeneous measurements with different frequency (Shan et al., 2020). By defining a Gaussian noise model, the framework is to find the minimized value by solving the nonlinear least-squares problem, as Equation 2.

$$T_B^{W*} = \operatorname{argmin} \sum_{k=0, K} \left( \|e_k^{LiDAR}\|_{\Sigma_k^{LiDAR}}^2 + \|e_{i,j}^{loop}\|_{\Sigma_{i,j}^{loop}}^2 + \|e_k^{ground}\|_{\Sigma_k^{ground}}^2 + \|e_n^{IMU}\|_{\Sigma_n^{IMU}}^2 + \|e_t^{matching}\|_{\Sigma_t^{matching}}^2 \right) \quad (2)$$

where  $e_k^{LiDAR}$ ,  $e_{i,j}^{loop}$ ,  $e_k^{ground}$ ,  $e_n^{IMU}$ , and  $e_t^{matching}$  denote the LiDAR odometry, loop closure, ground constraints, IMU pre-integration, and map matching constraint, respectively.  $K$  denotes the number of nodes in the factor graph. Each factor contributes to the pose estimation when a new state node is added to the graph. Then the factor graph is subsequently optimized based on the Bayes tree (iSAM2) using incremental smoothing and mapping framework (Kaess et al., 2011). The process for generating the aforementioned factors and constraints is described in the following sections.

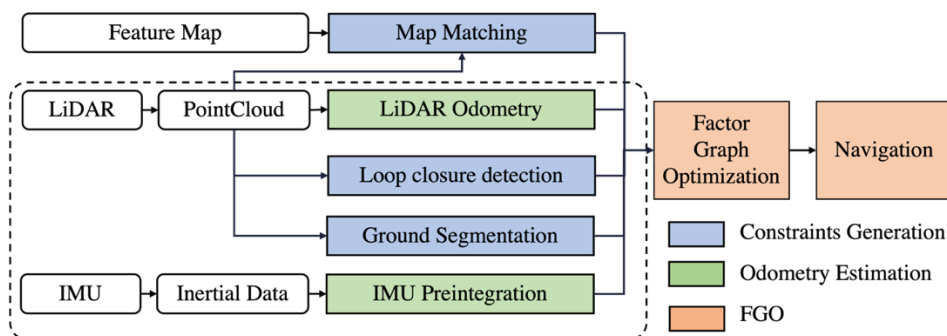


Figure 1. The pipeline of the proposed framework.

### 3.2 Tightly coupling LiDAR-IMU odometry

The tight coupling of LiDAR and IMU enables the estimation of the relative motion between each relative state node incorporating IMU pre-integration and LiDAR odometry. Details are described in the following sections.

#### 3.2.1 IMU pre-integration

Following the algorithm and solution from (Forster et al., 2017), the IMU pre-integration based on raw angular velocity and acceleration is defined, as shown in Equation 3 and Equation 4. The IMU pre-integration provides the relative transformation between two received IMU raw data and provides a state factor while mitigating the motion drift of point clouds.

$$\begin{aligned}\widehat{\omega}_t &= \omega_t + b_t^\omega + n_t^\omega \\ \widehat{a}_t &= R_t^{BW}(a_t - g) + b_t^a + n_t^a\end{aligned}\quad (3)$$

where  $\widehat{\omega}_t$  = raw angular velocity  
 $\widehat{a}_t$  = raw acceleration  
 $R_t^{BW}$  = transformation matrix from  $W$  to  $B$

$$\begin{aligned}\Delta v_{ij} &= R_i^T(v_j - v_i - g\Delta t_{ij}) \\ \Delta p_{ij} &= R_i^T(p_j - p_i - v_i\Delta t_{ij} - \frac{1}{2}g\Delta t_{ij}^2) \\ \Delta R_{ij} &= R_i^T R_j\end{aligned}\quad (4)$$

where  $\Delta v_{ij}$ ,  $\Delta p_{ij}$ ,  $\Delta R_{ij}$  represent the velocity, position, and rotation of the relative motion in the body frame.

The IMU mechanization provides highly frequent relative pose information through its measurements. Meanwhile, the IMU bias is jointly optimized using FGO during motion to mitigate the drift.

#### 3.2.2 LiDAR odometry estimation

When a new frame of point clouds is available, several steps are implemented to estimate the local LiDAR odometry, including motion drift calibration, feature extraction, scan matching and relative transformation estimation (J Zhang and Singh, 2014). Here, we provide a brief overview of the LiDAR odometry estimation process.

By integrating IMU data, the first step is to perform motion drift calibration to synchronize the timestamp of each frame. Next, edge and planar points are then extracted as featured point  $\{F_k^p, F_k^e\}$  by calculating and sorting the roughness of points. The distance between relative frames is minimized and the relative transformation is calculated. The process can be formulated as Equation 5,

$$m_{k-1,B}^{k,B} = \rho(F_k\{F_k^p, F_k^e\}, F_{k-1}\{F_{k-1}^p, F_{k-1}^e\}). \quad (5)$$

where  $\rho$  denotes the distance minimization operation.  $m_{k-1,B}^{k,B}$  demonstrates the coarse relative pose information between the feature point set  $F_k\{F_k^p, F_k^e\}$  and feature point set  $F_{k-1}\{F_{k-1}^p, F_{k-1}^e\}$ .

Two steps are involved in estimating the relative transformation matrix. First, the corresponding edge and planar points are registered by comparing the feature patches. Second, with an initial value is estimated from the IMU pre-integration, a Gauss-Newton method is used to solve the non-linear function for

minimizing the distance. The relative transformation matrix can be represented using Equation 6,

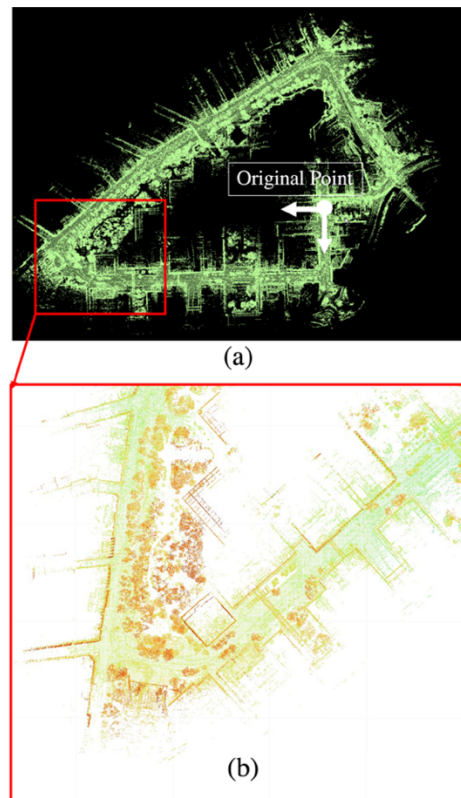
$$\Delta T_{k-1,k} = T_{k-1}^T T_k \quad (6)$$

where  $\Delta T_{k-1,k}$  is the relative LiDAR odometry between two states  $T_k$  and  $T_{k-1}$ .

### 3.3 Map matching

In this section, we provide a comprehensive explanation of incorporating map matching between LiDAR perception and the prior feature map for the localization task.

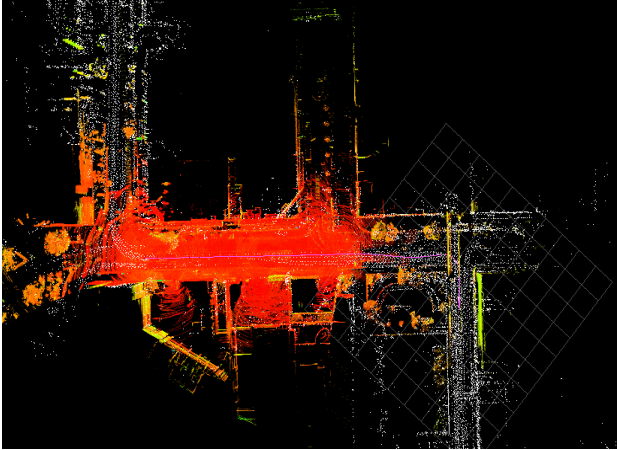
Prior to online localization, a prior feature map is generated to represent the geometric features of the surrounding structures. In general, we utilize the ground truth poses to align featured points extracted from LiDAR to construct the feature map along the trajectory. Firstly, pre-processing is performed, which includes coordinate transformation and time synchronization, between GNSS and LiDAR. This ensures that both data sources are properly aligned for further analysis. Secondly, a feature extraction module is designed to extract stable points from LiDAR. Specifically, ground points are segmented to provide consistent ground planes, while corner points and planar points are extracted to encode the geometric information of the environment. Additionally, a denoising algorithm is implemented to eliminate the effect of noise points and non-stationary objects. Finally, the offline map is generated by aligning continuous feature points using the ground truth trajectory. An example of the feature map is shown in Figure 2.



**Figure 2.** One example of prior feature map, where (a) represents the global map and (b) shows the zoomed-in version of the global map. The zoomed-in map clearly illustrates the boundary and consistent ground points.

The aim of the proposed map matching process is to establish relative transformation relationship between the onboard LiDAR and the prior feature map. In general, the distance error of the

map matching is expected to be zero, which can be an absolute constraint for the relative transformation estimation. To generate the map matching factor, the Iterative Closet Point (ICP) is implemented to align the corresponding points between onboarding LiDAR and the feature map. Here, to reduce the computation cost, sparse keyframes are selected with predefined distance and angle thresholds and are utilized to generate the constraint. An example of map matching is shown in Figure 3.



**Figure 3.** An example of map matching between received featured point clouds (coloured points) and the down sampled feature maps (white points) in the Linux system.

### 3.4 Ground constraint

This section outlines the approach employed to detect ground points and constraints from point clouds captured by the onboard LiDAR.

Various techniques have been proposed for extracting the ground plane from point clouds, and one widely used and efficient algorithm is Random Sample Consensus (RANSAC) (Ni et al., 2016). Prior to ground detection, points clustering is implemented via a pre-defined Euclidean distance and ground points segmentation is achieved by exploiting the angle of the received laser signals, which enables estimation of the relative height of the vehicles in relation to the ground. In this work, the ground plane model is defined using the normal vector of the detected surface and the Euclidean distance between the points to the detected ground surface, as follows,

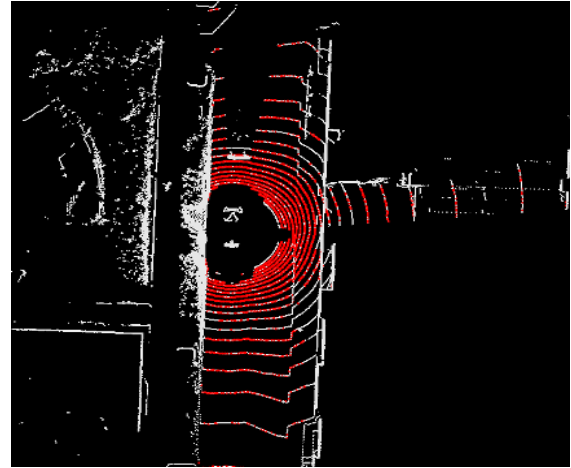
$$G_k = [n_k^x, n_k^y, n_k^z, d_k]^T \quad (7)$$

where  $(n_k^x, n_k^y, n_k^z)$  represents the normal vector and  $d_k$  is the Euclidean distance. Therefore, the process of ground detection via RANSAC can be defined as Equation 8,

$$n_k^* = \operatorname{argmin} \|n_k^x P_{x,g} + n_k^y P_{y,g} + n_k^z P_{z,g} - d_k\| \quad (8)$$

where  $n_k^*$  denotes the optimal parameters of the ground plane. The  $P_g = \{P_{x,g}, P_{y,g}, P_{z,g}\}$  denotes several points contributed to ground surface.

Figure 4 illustrates an example of ground detection using RANSAC in urban environments. The red points indicate the ground points, while the white points represent the non-ground points. It is worth noting that the algorithm is capable of accurately detecting and labelling the ground points, even in the presence of dynamic objects such as vehicles on the road. By assuming the ground model is consistent as plane primitive along with the trajectory, the difference of detected ground planes can be considered as ground constraints and optimized in the FGO.



**Figure 4.** Examples of ground points segmentation (raw data: white points, segmented ground: red points) in the Urbannav dataset.

### 3.5 Loop detection

To perform global constraints, loop closure detection is a useful technique to eliminate the accumulative errors. It involves comparing the current frame with historical frames and applying the global constraint. To efficiently identify potential loop closures, this paper employs the scan context (SC) algorithm for loop candidate selection. When the difference of SC is less than a threshold  $\tau$ , the loop closure is accepted when the distance is less than distance threshold  $d_{lc}$ . A brief description of the SC descriptor and loop closure is introduced as follows:

The (SC) algorithm serves as a global descriptor for place recognition and facilitates the descriptor of loop closures (Kim and Kim, 2018). The SC descriptor divides a frame point clouds into azimuthal bins and radial bins, with the number of sectors  $N_s$  and rings  $N_r$ . The unorganized point clouds can be transformed using Equation 9.

$$P = \bigcup_{i \in [N_r], j \in [N_s]} P_{i,j} \quad (9)$$

where  $P =$  point clouds  
 $[i, j] =$   $i^{\text{th}}$  ring and  $j^{\text{th}}$  sector

Then the value of each bin of matrix is defined as the maximum value of the height value, as the following Equation 10,

$$I = \phi(P_{i,j}) \quad (10)$$

where  $\phi(P_{i,j}) = \max_{p \in P_{i,j}} z(p)$  demonstrates the operation that return the matrix value of  $z$  direction.

Given SC pairs, the similarity between two SC descriptor is calculated as Equation 11,

$$c^* = \operatorname{argmin}_{c_k \in \mathcal{C}} D(I^q, I^{c_k}), \text{ s.t } D < \tau \quad (11)$$

where  $D(I^q, I^{c_k}) =$  Euclidean distance between the two SC pairs.

$\tau =$  predefined threshold  
 $c^* =$  index of loop closure

After comparing the SC distances, the loop closure is selected by searching the closest distance of feature points in Euclidean space. The relative transformation matrix between the accepted loop is considered as loop closure factor, following Equation 12,



$$T_{i,j}^{loop} = (p_{i,j}^{loop}, q_{i,j}^{loop}) \quad (12)$$

where  $T_{i,j}^{loop}$  denotes the transformation matrix between the detected frames.  $p_{i,j}^{loop}$  and  $q_{i,j}^{loop}$  represent the translation and quaternion form, respectively.

### 3.6 GNSS localization (optional)

Despite the ability of map matching can provide reliable geo-reference and constraints, accumulative errors may affect the performance of localization. To address this issue, a GNSS system is considered as an optional solution to provide absolute measurements.

When GPS signal is received, it is transformed into the Cartesian coordinate using the Monarto projection. Subsequently, an initialization process is carried out to evaluate the accuracy of the current GPS signal by comparing the position between the state of the last node and the current GPS position. A new GPS factor is then added as a new node into the factor graph. In addition, to synchronize with other factors, linear interpolation is employed among the GPS measurements based on timestamps.

### 3.7 Factor graph optimization

This section provides an overview of the construction and optimization process of the factor graph optimization, incorporating the designed constraints and the motion state. The factor graph efficiently integrates the constraints, including map matching constraints and ground constraints, with the global constraints of loop closure, as well as the LiDAR odometry and IMU pre-integration. The structure of the factor graph is shown in Figure 5, which includes four types of constraints and the state node of the localization in the  $W$ . The state node is represented using  $T$  which is defined in Equation 1, which encodes the pose information along the trajectory.

The LiDAR odometry factor  $e_{k-1 \rightarrow k}^{LiDAR}$  and IMU pre-integration factor  $e_{k-1 \rightarrow k}^{IMU}$  can be represented as Equation 13 and Equation 14, respectively, which are calculated based on odometry estimation,

$$e_{k-1 \rightarrow k}^{LiDAR} = \left\| \left( T_{B,k-1}^W \quad -1 T_{B,k}^W \right), \left( T_{B,k-1}^L \quad -1 T_{B,k}^L \right) \right\|_{\Sigma_k^{LiDAR}}^2 \quad (13)$$

$$e_{k-1 \rightarrow k}^{IMU} = \left\| T_{B,k-1}^W \quad -1 T_{B,k}^W \right\|^2 \quad (14)$$

where  $\Sigma_k^{LiDAR}$  and  $\Sigma_k^{IMU}$  denote the information matrix of LiDAR and IMU odometry, respectively.

The loop closure constraint is defined by the Euclidean distance between the detected loop as following,

$$e_{i,j}^{loop} = \left\| \left( T_{L,k-1}^W \quad -1 T_{L,k}^W \right), \left( T_{L,i,j}^L \quad -1 T_{L,i}^L \right) \right\|_{\Sigma_k^{loop}}^2 \quad (15)$$

Given that the ground constrain is based on the detected ground plane, the  $e_k^{ground}$  can be represented using the normal vector, as demonstrated in Equation 16 (Wen and Hsu, 2022).

$$e_{k \rightarrow k+1}^{ground} = \|n_k^W - n_0^W\|^2 \quad (16)$$

where the  $n_0^W = [0 \ 0 \ 0 \ 1]$  is expected as the assumption of flat ground.

As for the map matching constraint, the error function can be formulated as,

$$e_{k \rightarrow k+1}^{map\ matching} = \left\| T_{B,k-1}^W \quad -1 T_{B,k}^W \right\|^2 \quad (17)$$

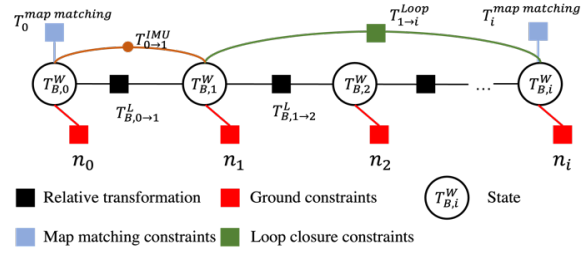


Figure 5. The structure of factor graph optimization.

## 4. EXPERIMENTS

To evaluate the proposed framework, we conducted experiments using an opensource dataset from UrbanNav (Hongkong) with medium density is selected (Hsu et al., 2021). The detailed description of the dataset is shown in Table 1. Meanwhile, some key parameters used in experiments are listed in Table 2. During the evaluation, we compared to the state-of-art Lego-LOAM and NDT-based solution algorithms (Kato et al., 2018; Shan and Englot, 2018). The accuracy evaluation for translation and rotation is defined as following. Meanwhile, we implement the evaluating toolkit to calculate the relative error (RPE) for the translation and rotation in movement direction, including x axis and y-axis (Grupp, 2017).

- RMSE: standard deviation of the trajectory
- MEAN: Mean error of the trajectory

Table 1. Dataset description.

Description	UrbanNav (Medium Density)
IMU sensor	Xsens Mti 10, 400 Hz, AHRS
LiDAR scanner	32 (beams)
Trajectory length	3.64 (km)
Number of Scans	7848 (frames)
Dynamics	With various moving objects

Table 2. Parameters in the experimental evaluation

Parameters	Value	Description
$\delta_\theta$	1 m	Keyframe selection
$\delta_t$	15°	Keyframe selection
$\sigma$	0.2	Down sample rate
$\tau$	0.5	Scan Context

Table 3. Accuracy evaluation in translation.

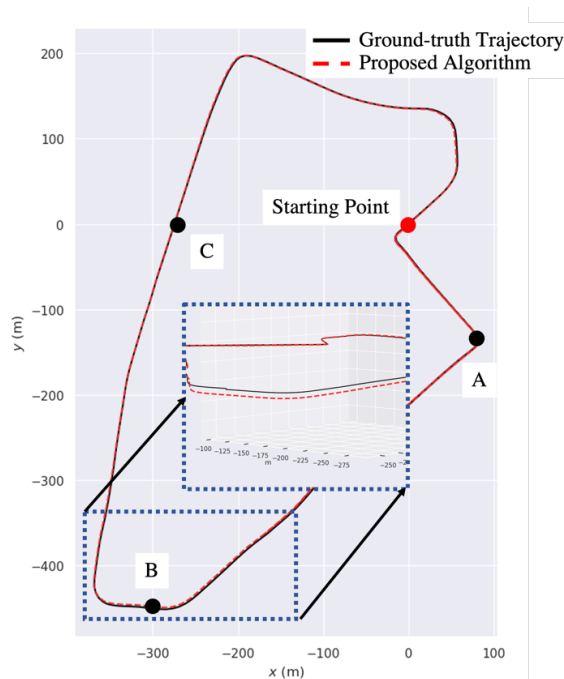
Accuracy	Algorithms (m)		
	Lego-LOAM	NDT-based method	Our algorithm
MEAN	19.46	5.72	1.07
RMSE	50.00	16.10	2.29

As shown in Figure 6, the proposed solution successfully estimates a highly accurate trajectory compared to the ground-truth pose information. Specifically, the trajectory obtained by Lego-LOAM is highly drifted as the performance of LiDAR odometry is easily affected in dynamic environments with many moving objects. Additionally, the global optimization of Lego-LOAM is only based on loop closure in the graph optimization algorithm, and large drift has been accumulated before the loop

is obtained along the trajectory. As for the NDT-based algorithm, the performance of the trajectory depends only on the accuracy of NDT matching, which is easily affected by moving objects. Compared to these two algorithms, our proposed algorithm achieves more stable results in urban canyon environments.

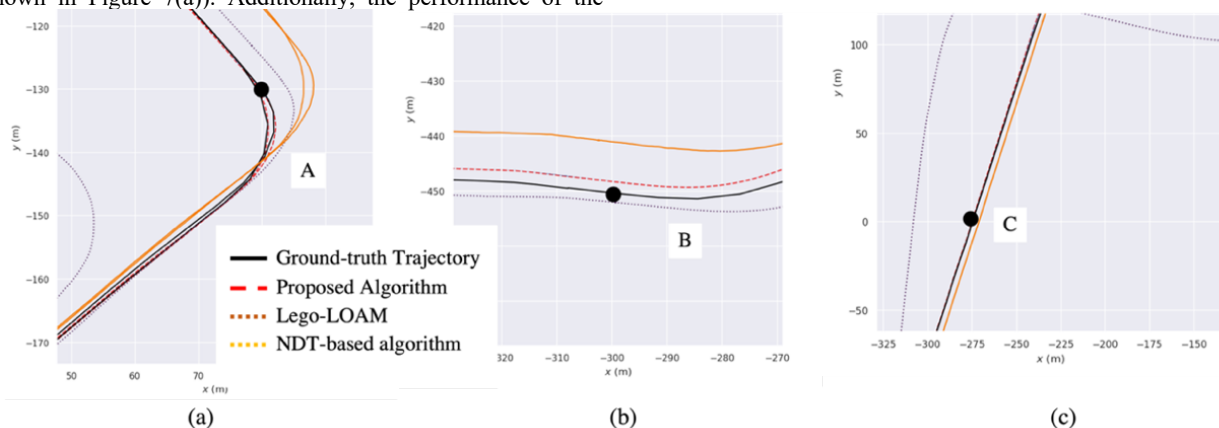
**Table 4.** Accuracy evaluation in rotation.

Accuracy	Algorithms (degree)		
	Lego-LOAM	NDT-based method	Our algorithm
MEAN	12.87	8.13	3.18
RMSE	29.36	15.36	5.39



**Figure 6.** Estimated trajectory using our proposed method and ground-truth trajectory.

Figure 7 shows the detailed and zoomed-in results of the estimated trajectories using different algorithms. We selected turning (location A), after-turning (location B), and long-term navigation (location C) for accuracy comparison. When the vehicle enters the turning, the result of Lego-LOAM is highly affected as the overlapping between relative frames decreases (as shown in Figure 7(a)). Additionally, the performance of the



**Figure 7.** Evaluation results compared to ground truth trajectory.

NDT-based algorithm continuously decreases as errors accumulate via point cloud registration. Therefore, our proposed algorithm achieves stable performance by enhancing error elimination and local constraints.

The error evaluation of rotation and translation along the trajectory are shown in Table 3 and Table 4. In general, our proposed algorithm achieves more stable results than Lego-LOAM and NDT-based algorithm. Regarding the mean rotation error, the proposed algorithm achieves around meters error in the long-term navigation task. With the help of IMU preintegration, the trajectory is stable even where there is a turning (as shown in Figure 7).

## 5. CONCLUSION AND FUTURE WORK

This paper presents a localization framework based on LiDAR and inertial measurements, aiming to enhance robustness in urban canyon environments. The proposed approach formulates localization within a FGO framework, utilizing state nodes and constraints. Specifically, LiDAR odometry and IMU preintegration estimate the relative transformation to link the relative state node, while three separate constraints in the form of factors are introduced through segmented ground constraint, map matching, and loop closure. The proposed algorithm offers two key advantages: Firstly, by employing the FGO framework, heterogeneous measurements can be fused, effectively mitigating cumulative errors. Secondly, the framework can efficiently enhance positioning performance at two stages of odometry estimation and global optimization by fusing different constraints. Experimental results demonstrate that the navigation solution can achieve stable and high accuracy positioning in urban canyon environments.

Future work will concentrate on the feature representation and association to improve the accuracy of registration and place recognition. The limitation of our work is that we utilize all the point clouds obtained from LiDAR. For the process of point registration, moving objects (e.g., moving cars) will bring in large error feature association and the non-rigid transformation matrix between relative state nodes. Feature work will focus on removing moving objects based on semantics and enhancing the positioning in dense environments.

## ACKNOWLEDGEMENTS

This research was supported by funding provided to Professor Naser El-Sheimy from NSERC CREATE and Canada Research Chairs programs.

## REFERENCES

- Ai, M., Luo, Y., El-Sheimy, N., 2022. Surround Mask Aiding GNSS/LiDAR SLAM for 3D Mapping in the Dense Urban Environment. Proc. 35th Int. Tech. Meet. Satell. Div. Inst. Navig. (ION GNSS+ 2022) 2011–2019. <https://doi.org/10.33012/2022.18550>
- Bársan, I.A., Wang, S., Pokrovsky, A., Urtasun, R., 2020. Learning to Localize Using a LiDAR Intensity Map.
- Chen, X., Läbe, T., Milioto, A., Röhling, T., Behley, J., Stachniss, C., 2022. overlapNet: a siamese network for computing LiDAR scan similarity with applications to loop closing and localization. *Auton. Robots* 46, 61–81. <https://doi.org/10.1007/s10514-021-09999-0>
- Cho, Y., Kim, G., Lee, S., Ryu, J.-H., 2022. OpenStreetMap-Based LiDAR Global Localization in Urban Environment Without a Prior LiDAR Map. *IEEE Robot. Autom. Lett.* 7, 4999–5006. <https://doi.org/10.1109/lra.2022.3152476>
- Dong, H., Chen, X., Stachniss, C., 2021. Online range image-based pole extractor for long-term LiDAR localization in urban environments. 2021 10th Eur. Conf. Mob. Robot. ECMR 2021 - Proc. <https://doi.org/10.1109/ECMR50962.2021.9568850>
- Dube, R., Dugas, D., Stumm, E., Nieto, J., Siegart, R., Cadena, C., 2017. SegMatch: Segment based place recognition in 3D point clouds. Proc. - IEEE Int. Conf. Robot. Autom. 5266–5272. <https://doi.org/10.1109/ICRA.2017.7989618>
- Forster, C., Carlone, L., Dellaert, F., Scaramuzza, D., 2017. On-Manifold Preintegration for Real-Time. *IEEE Trans. Robot.* 33, 1–21.
- Hsu, L.-T., Kubo, N., Wen, W., Chen, W., Liu, Z., Suzuki, T., Meguro, J., 2021. UrbanNav: An open-sourced multisensory dataset for benchmarking positioning algorithms designed for urban areas, in: Proceedings of the 34th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2021). pp. 226–256.
- Imanullah, M., Yuniarno, E.M., Sumpeno, S., 2019. SIFT and ICP in Multi-view based Point Clouds Registration for Indoor and Outdoor Scene Reconstruction. Proc. - 2019 Int. Semin. Intell. Technol. Its Appl. ISITIA 2019 288–293. <https://doi.org/10.1109/ISITIA.2019.8937292>
- Kaess, M., Johannsson, H., Roberts, R., Ila, V., Leonard, J.J., Dellaert, F., 2011. iSAM2: Incremental smoothing and mapping using the Bayes tree. *Int. J. Rob. Res.* 31, 216–235. <https://doi.org/10.1177/0278364911430419>
- Kato, S., Tokunaga, S., Maruyama, Y., Maeda, S., Hirabayashi, M., Kitsukawa, Y., Monroy, A., Ando, T., Fujii, Y., Azumi, T., 2018. Autoware on Board: Enabling Autonomous Vehicles with Embedded Systems. Proc. - 9th ACM/IEEE Int. Conf. Cyber-Physical Syst. ICCPS 2018 287–296. <https://doi.org/10.1109/ICCPS.2018.00035>
- Kim, G., Kim, A., 2018. Scan Context: Egocentric Spatial Descriptor for Place Recognition Within 3D Point Cloud Map. *IEEE Int. Conf. Intell. Robot. Syst.* 4802–4809. <https://doi.org/10.1109/IROS.2018.8593953>
- Levinson, J., Thrun, S., 2010. Robust vehicle localization in urban environments using probabilistic maps. Proc. - IEEE Int. Conf. Robot. Autom. 4372–4378. <https://doi.org/10.1109/ROBOT.2010.5509700>
- Liu, H., Ye, Q., Wang, H., Chen, L., Yang, J., 2019. A precise and robust segmentation-based lidar localization system for automated urban driving. *Remote Sens.* 11, 1–18. <https://doi.org/10.3390/rs11111348>
- Liu, X., Zhang, L., Qin, S., Tian, D., Ouyang, S., Chen, C., 2019. Optimized loam using ground plane constraints and segmatch-based loop detection. *Sensors (Switzerland)* 19. <https://doi.org/10.3390/s19245419>
- Liu, Y., Wang, C., Wu, H., Wei, Y., Ren, M., Zhao, C., 2022. Improved LiDAR Localization Method for Mobile Robots Based on Multi-Sensing. *Remote Sens.* 14. <https://doi.org/10.3390/rs14236133>
- Lu, W., Wan, G., Zhou, Y., Fu, X., Yuan, P., Song, S., 2019. DeepICP: An End-to-End Deep Neural Network for 3D Point Cloud Registration.
- Ni, H., Lin, X., Ning, X., Zhang, J., 2016. Edge detection and feature line tracing in 3D-point clouds by analyzing geometric properties of neighborhoods. *Remote Sens.* 8. <https://doi.org/10.3390/rs8090710>
- Nüchter, A., Bleier, M., Schauer, J., n.d. Continuous-Time SLAM — Improving Google’s Cartographer 3D Mapping 53–73.
- Rozenberszki, D., Majdik, A., 2020. LOL: Lidar-Only Odometry and Localization in 3D Point Cloud Maps.
- Schaefer, A., Buscher, D., Vertens, J., Luft, L., Burgard, W., 2019. Long-term urban vehicle localization using pole landmarks extracted from 3-D lidar scans. 2019 Eur. Conf. Mob. Robot. ECMR 2019 - Proc. <https://doi.org/10.1109/ECMR.2019.8870928>
- Shan, T., Englot, B., 2018. LeGO-LOAM: Lightweight and Ground-Optimized Lidar Odometry and Mapping on Variable Terrain, in: IEEE International Conference on Intelligent Robots and Systems. pp. 4758–4765. <https://doi.org/10.1109/IROS.2018.8594299>
- Shan, T., Englot, B., Meyers, D., Wang, W., Ratti, C., Rus, D., 2020. LIO-SAM: Tightly-coupled lidar inertial odometry via smoothing and mapping. *IEEE Int. Conf. Intell. Robot. Syst.* 5135–5142. <https://doi.org/10.1109/IROS45743.2020.9341176>
- Wen, W., Hsu, L.T., 2022. AGPC-SLAM: Absolute Ground Plane Constrained 3D Lidar SLAM. *Navig. J. Inst. Navig.* 69. <https://doi.org/10.33012/navi.527>
- Zhang, Ji, Singh, S., 2014. LOAM: Lidar Odometry and Mapping in Real-time, in: Robotics: Science and Systems X. <https://doi.org/10.15607/RSS.2014.X.007>
- Zhang, J, Singh, S., 2014. LOAM: Lidar Odometry and Mapping in Real-time, Robotics: Science and Systems, . *Robot. Syst.* 2, 9–17.
- Zhu, Y., Ma, Y., Chen, L., Liu, C., Ye, M., Li, L., 2020. GOSMatch: Graph-of-semantics matching for detecting loop closures in 3D LiDAR data. *IEEE Int. Conf. Intell. Robot. Syst.* 5151–5157. <https://doi.org/10.1109/IROS45743.2020.9341299>