LIDAR-INERTIAL NAVIGATION BASED ON MAP AIDED DISTANCE CONSTRAINT AND FACTOR GRAPH OPTIMIZATION

Mengchi Ai[1], Mohamed Elhabiby[2], Ilyar Asl Sabbaghian Hokmabadi[1], Naser El-Sheimy[1*]

[1]Dept. of Geomatics Engineering, University of Calgary, 2500 University Dr NW, Calgary, AB T2N 1N4, Canada – (mengchi.ai, ilyar.aslsabbaghianh, elsheimy) @ucalgary.ca
[2]Micro Engineering Tech. Inc., Calgary, AB, Canada, elhabiby@microengineering.ca

KEY WORDS: Automatic driving, LiDAR-IMU integration, Sensor fusion, Factor graph optimization, OpenStreetMap, map aided navigation

ABSTRACT:
The simultaneous localization and mapping (SLAM) is one of the well-developed positioning technology that provides high accuracy and reliability positioning for automatic vehicles and robotics applications. Integrating Light Detection and Ranging (LiDAR) with an Inertial Measurement Unit (IMU) has emerged as a promising technique for achieving stable navigation results in dense urban environments, outperforming vision-based or pure Inertial Navigation System (INS) solutions. However, conventional LiDAR-Inertial SLAM systems often suffer from limited perception of surrounding geometric information, resulting in unexpected and accumulating errors. In this paper, we proposed a LiDAR-Inertial SLAM scheme that utilizes a prior structural information map which is generated from opensource OpenStreetMap (OSM). In contrast to conventional solutions of OSM-aided SLAM approaches, our method extracts the vectorized models of road and building and synthetically generates dense point maps for LiDAR registration. Specifically, a structural map processing module extracts the road models and building models from OSM and generates a structure information map (SIM) with dense point clouds. Secondly, a map aided distance (MD) constraint is calculated by registering selected keyframes and the prior SIM. Finally, a factor graph optimization (FGO) algorithm is involved to integrate the relative transformation obtained from LiDAR odometry, IMU pre-integration, and the map aided distance constraints. To evaluate the proposed LiDAR-based positioning accuracy, experimental evaluation is implemented in an opensource dataset collected in the urban canyon environments. Experimental results demonstrates that with the help of the proposed MD constraint, the LiDAR-based navigation solution can achieve accurate positioning, with a root mean square deviation (RSME) of 4.7 m.

1. INTRODUCTION

Accurate and reliable navigation is a crucial requirement for autonomous driving, providing real-time localization information for path planning, vehicle control, and decision making (Ai et al., 2022). While the fusion of onboard sensors has obtained reasonably accurate position in most types of urban environments, leveraging static urban structural information and traffic data, particularly high-definition (HD) maps, can further enhance the accuracy in urban canyon environments. Consequently, integrating an HD map with a fused multi-sensor system has emerged as a critical technique for advancing autonomous driving beyond Level-3.

Recent studies on HD map-aided navigation have primarily focused on utilizing HD maps to provide static geo-referencing traffic information and enhance localization performance. These studies can be classified into two groups based on the representation of HD maps: model-based algorithms and point-based algorithms. Model-based algorithms utilize vectorized HD maps (e.g., Opendrive) to provide landmarks and traffic shapes as constraints, improving local odometry estimation (Bender et al., 2014; Pai et al., 2022). While traffic model can encode the traffic in a mathematical representation, it's performance of corresponding association for place recognition is limited due to the sparsity of the parametric model. In contrast, point-based algorithms involve point cloud maps that consist of rich semantic and geometric points (e.g., Lanelet map) and can be registered with onboard LiDAR using registration algorithms, such as normal distributions transform (NDT) and iterative closest point (ICP) (Bender et al., 2014). Inspired by these solutions, to enhance the positioning in urban canyon environments, this work incorporates a point map-aided solution to provide matching constraints.

Despite the advancements in current map-aided navigation solutions, there remain challenging issues which can be discussed in two aspects. Firstly, the construction of point maps relies on extensive manual work and data collection. A mobile vehicle equipped with LiDAR and INS should scan and register each local area along the trajectory based on a high accurate trajectory. This process becomes particularly challenging in urban canyon environments where there are plenty of moving objects. Secondly, current solutions primarily formulate the map matching as an additional constraint, which bringing in unexpected registration errors into the localization system. Consequently, achieving accurate positioning in urban canyon environments still surfers from the challenging issues of the construction and utilization of point clouds map.

To address the issues, we propose a localization framework that combines the opensource OSM and the integration of LiDAR-Inertial system. Prior to the navigation task, the proposed framework incorporates a structural map processing module that generates a structural information map (SIM) from the vectorized road and building models which are available in OSM. During the localization stage, the initialization involves providing a manually provided coarse positioning and refining the positioning by aligning the LiDAR and the SIM. In the tracking stage, the relative trajectory is estimated based on LiDAR odometry estimation, IMU pre-integration, and the absolute distance between the LiDAR and SIM. Finally, all the relative poses are integrated based on a factor graph optimization (FGO) and the global trajectory is estimated by solving the non-linear function using iSAM2 (Kaess et al., 2011).
The contributions of the proposed method can be summarized as follows:

- A map-aided LiDAR-Inertial navigation framework is proposed to integrate the relative positioning from LiDAR odometry, IMU pre-integration and the map aided absolute constraints.

- A structural map processing module is introduced to generate a structural information map that is composed of synthetic points from opensource OSM.
- An absolute distance constraint based on map matching is proposed to enhance the performance of map-aided LiDAR -Inertial system.

2. RELATED WORKS

The map aided LiDAR-based navigation aims to estimate the relative pose between consecutive frame and IMU measurements while integrating the positioning based on the alignment between the onboard sensor and the prior map. This section provides a brief review of map-aided LiDAR-based navigation, focusing on two aspects: the algorithms of feature association, and the optimization model for odometry estimation.

To align the onboard LiDAR with the prior global map, there are two forks of research that focuses on alignment of point clouds: registration-based algorithms, and feature-based algorithms (Li et al., 2022). The registration-based algorithms involve dense point clouds map as the prior map to provide the geometric information of the surrounding environments. For example, in (Chiang et al., 2023), initial position is estimated based on the NDT alignment between the point clouds and dense point clouds map. In (Gui et al., 2022), NDT is implemented to align the LiDAR to a pre-built offline map to estimate the relative pose along the trajectory and the map matching constraints is to be optimized in the factor graph. These registration-based algorithms can estimate stable relative positioning but with a limited computational efficiency. On the other hand, feature-based algorithms extract geometric and semantic features and register the corresponding features with the prior LiDAR maps. For example, in (Chen et al., 2019; Liu et al., 2020; Schaefer et al., 2019), semantic patches (e.g., pole like objects) are considered as stable points to provide environmental context for localization tasks. In (Cho et al., 2022; Yan et al., 2019), feature spaces consist of semantic descriptors generated from the vectorized models of roads and building boundaries, representing the distribution of traffic elements. However, the performance of feature description may be affected due to the dynamic objects and seasonal variation. Inspired by the utilization of OSM, we propose a point-based SIM to represent the environmental context including the road shape and building boundaries.

To integrate the relative positioning estimated based on the alignment with the prior map, and other heterogenous measurements, researchers have formulated navigation as an optimization problem or employed filter-based frameworks. In (Elhousni et al., 2022), Monte Carlo localization solution, as known as a particle filter solution, is implemented to estimate the trajectory based on a designed cross-modal localization framework. In (Liu et al., 2019), the Kalman filter is involved to estimate the position, velocity, as well as an attitude which is calculated based on map alignment. For optimization-based solutions, error functions integrated distance errors are designed to mitigate accumulative errors when constraints are available. For example, in (Frosi et al., 2023), the pose graph optimizer is designed to fuse the relative pose estimated from local pose trackers. With the advantage of efficient data fusion, a factor graph optimization is selected to fuse the LiDAR odometry, IMU measurement, and the map matching constraints in this paper.

3. OVERVIEW

In this section, we will introduce an overview pipeline of the proposed navigation algorithm and the predefined coordinate and the notations.

To make the proposed framework clearer, the notations and coordinate definition are used by the rest of this paper: The local world frame $W\{X^W, Y^W, Z^W\}$ is defined as a world coordinate which is fixed with the original start point of the vehicle. The SIM $X_s\{x_s^W, y_s^W, z_s^W\}$ obtained from OSM is transformed into W using the start point as the reference point. The LiDAR frame $L\{X^L, Y^L, Z^L\}$ denote the LiDAR coordinate at each LiDAR frame and is fixed with the LiDAR odometry coordinate. The vehicle body frame B is fixed with IMU coordinate. The transformation matrix T_B^L between L and B is pre-calibrated in a calibrated system. The coordinate definition is shown in Figure 1.

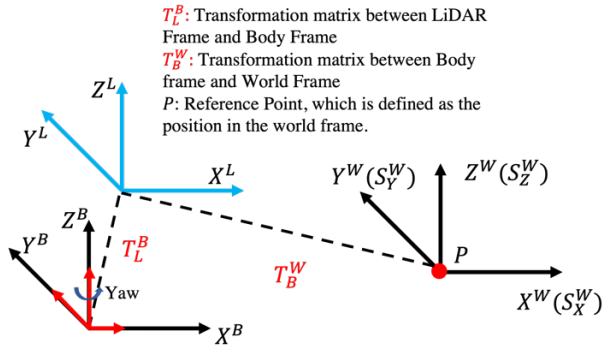

Figure 1. The coordinate definition.

The overview of the proposed navigation framework is shown in Figure 2. Assuming a vehicle is equipped with an onboard LiDAR and an IMU, raw point clouds and raw IMU measurements are continuously captured from the sensors with synchronized timestamps. Prior to the online navigation task, the proposed structural information map is generated using the road and building models from OSM. During the initialization process, a coarse position is manually provided for the fine alignment with SIM X_s using Iterative Closest Point (ICP) algorithm. For the tracking stage, relative pose is estimated based on LiDAR odometry estimation and the IMU pre-integration. In addition, the absolute distance constraint is generated during the trajectory by aligning the keyframes from LiDAR and the SIM, which represents the main contribution. Finally, the relative pose and the map matching constraint are integrated using an FGO, which enhance the fusion of heterogeneous measurements and constraints.

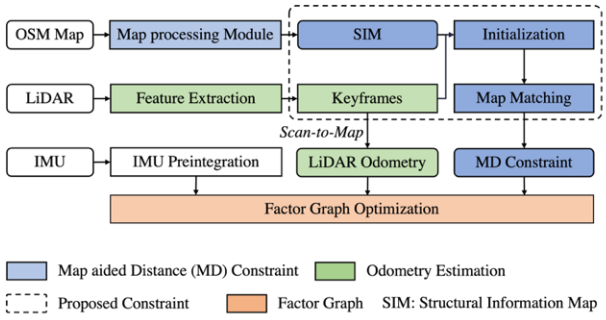

Figure 2. The workflow of the proposed algorithm.

The structure of the FGO is shown in Figure 3. The factor graph consists of three factors and one variable, which are used to model the relative transformation obtained from LiDAR odometry estimation, IMU pre-integration, and the map-aided distance constraint. The vehicle state, represented as node x, can be described using Equation 1,

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

where R = rotation matrix
p = position vector
v = speed
b = IMU bias

The estimation of the robot state is formulated as a maximum a posteriori (MAP) problem. To model this problem, a factor graph is used to model the MAP problem and the solution is obtained through a nonlinear least squares problem (Shan et al., 2020). The functional equation for constructing the factor graph consists of three types of factors and one variable type, as described in Equation 2,

$$\mathrm{T}_B^{W^*} = \operatorname{argmin} \sum_{k=0,K} \left(\left\| \mathrm{e}_k^{LiDAR} \right\|^2_{\Sigma_k^{\mathrm{LDAR}}} + \left\| \mathrm{e}_t^{IMU} \right\|^2_{\Sigma_t^{\mathrm{IMU}}} + \left\| \mathrm{e}_i^{MD} \right\|^2_{\Sigma_i^{\mathrm{MD}}} \right) \quad (2)$$

where the $\mathrm{T}_B^{W^*}$ denotes the estimated trajectory, which can be represented as $T_B^W = [R|p]$. e_k^{LiDAR}, e_t^{IMU}, and e_i^{MD} denotes the relative pose estimated from LiDAR, IMU and map aided distance constraints, while Σ_k^{LDAR}, Σ_t^{IMU}, and Σ_i^{MD} denotes the information matrix, respectively. K denotes the number of nodes that will be optimized within the factor graph.

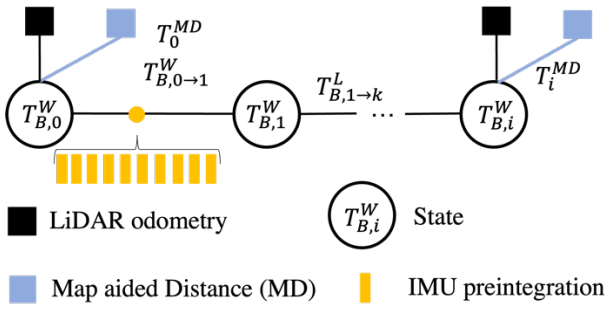

Figure 3. The structure of factor graph optimization.

4. METHODOLOGY

This section introduces the generation of the proposed SIM, and factors generation including the LiDAR odometry estimation, IMU pre-integration, and the map aided distance constraint.

4.1 Structural map processing module

Before the navigation task, the structural map processing module is introduced to generate the point map based on OSM. To make the implementation clearer, a brief introduction of the OSM is presented firstly. The OSM provides five layers of data for a selected area, including: strings, points, lines, multiline polygons, and the spatial relationship. In our approach, we extract the vectorized lines of the building boundary and the line points as raw data to represent the structural information. Figure 4 shows an example of the visualization of raw OSM points.

For the road shape, a Delaunay mesh is firstly implemented to estimate the shape of the building. Then an interpolation process is implemented to generate the road points within the obtained mesh and a predefined density, σ. As for road points, a linear interpolation function is implemented to generate the dense points around a centre point on the road, following the Equation 3. An example of the SIM map and the original OSM data is shown in Figure 5.

$$d_n^* \leq \Delta d + d_n, \theta_n \leq \Delta\theta + \theta_2 \quad (3)$$

Where Δd, $\Delta\theta$ denotes the interpolated value of a raw centre road point.

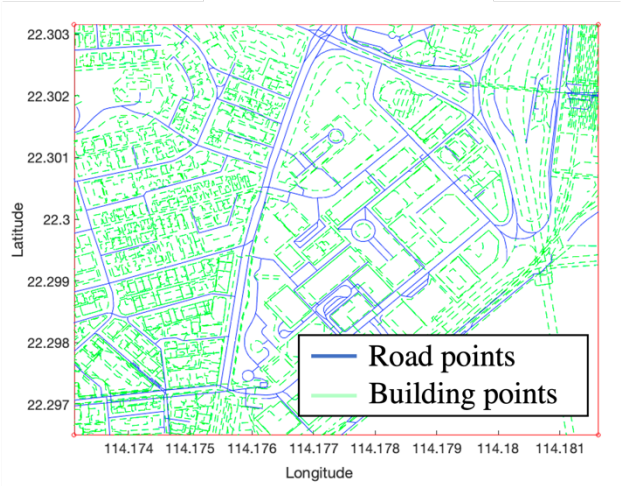

Figure 4. An example of raw OSM map. Road points and buildings points are coloured in blue and green, respectively.

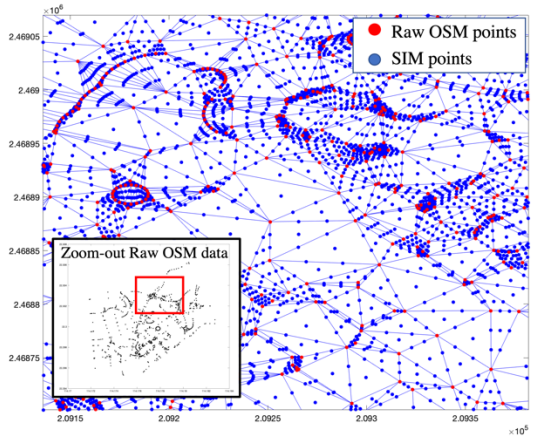

Figure 5. The zoom-in visualization of SIM of a selected area. (a) and (b) shows the raw data (blue points) and synthetic data (red points). In this example, the density of the interpolation σ=3.

As OSM is an open-source dataset, there may be discrepancies between the raw OSM data and the observation from the dataset. To address this issue, a consistent evaluation and alignment are implemented to assess the similarity between the ground-truth point cloud map and the generated SIM. Specifically, the ground-truth point map is generated by aligning the LiDAR frames using the ground-truth trajectory and manual labelling. Then an ICP is implemented to align the SIM and the generated point map. As the OSM only provide latitude and altitude information of the selected area, the vertical information and the building information are obtained from by extracting the shape of the building from OSM building and semantic point clouds. In addition, a downsampling rate is predefined to balance the density of the SIM.

4.2 LiDAR odometry estimation

The goal of LiDAR odometry estimation model extracts the feature points from raw point clouds, perform the feature association, and estimate the relative pose from the consecutive frames. In this paper, three steps are implemented to estimate the relative motion, including feature extraction, scan-to-map

matching, and LiDAR mapping (Zhang and Singh, 2014). A brief introduction is presented in this section.

For each new point clouds P in LiDAR frame L, feature points are firstly extracted using a design smooth value, which is calculated using the following Equation 4,

$$s_{i,j} = \frac{1}{\|S\|\|p_i\|}\left\|\sum_{j=1,j\neq i}^{S}(p_j - p_i)\right\| \quad (4)$$

where S is the surrounding sub-region around the center point p_i. By calculating all the smooth value, the feature points including corner points and planar points are extracted by sorting all the smooth value. When the smooth value is larger than a threshold, the point is selected as a corner point, otherwise, the point is labelled as a planar point. As a result, the output of the feature extraction is the feature point set, $F^L\{F_e^L, F_p^L\}$.

The relative pose between relative frames is estimated using the process of scan-to-map, which is estimated by minimizing the distance of feature points between the feature points $F^L\{F_e^L, F_p^L\}$ and the map M^W. The map $M$is generated by transforming the feature points set using the transformation matrix $T_i^{L\to W}$. Then the distance of point-to-plane and edge-to-plane is designed as the error function to estimate the relative pose between, as Equation 5,

$$\min_T(\, d('F_e^L, F_e^L), d('F_p^L, F_p^L)) \quad (5)$$

where the $'F_e^L$ and $'F_p^L$ are the transformed points in coordinate W. To solve this error function, the Gauss-Newton is used to solve the non-linear function to calculate the transformation matrix.

By minimizing the distance between relative frames, the transformation matrix between two states can be represented as the Equation 6,

$$\Delta T_{i,i+1} = T_i^T T_{i+1} \quad (6)$$

where $\Delta T_{i,i+1}$ denotes the relative transformation between state nodes. To eliminate the computation cost of factor graph optimization, keyframes are selected based on the motion distance and angle. Only relative transformation between keyframes will be optimized in the FGO. As a result, the LiDAR odometry factor can be represented as follows,

$$\left\|\mathrm{e}_k^{LiDAR}\right\|^2_{\Sigma_k^{\mathrm{LiDAR}}} = \left\|f\left(\left(\mathbf{T}_{B,k}^{W\ -1}\boldsymbol{T}_{B,k+1}^{W}\right),\left(\mathbf{T}_{B,k}^{L\ -1}\boldsymbol{T}_{B,k+1}^{L}\right)\right)\right\|^2_{\boldsymbol{\Sigma}_k^{LiDAR}} \quad (7)$$

where e_k^{LiDAR} denotes the information matrix of the error function. f denotes the minor operator.

4.3 Map matching aided distance constraints

In this section, the integration of the generated SIM and the LiDAR-Inertial system is introduced. There are two stages that integrated with the map matching constraint. Firstly, for the stage of initialization, a fine matching is implemented to estimate the accurate initial position based on a manual positioning. Note that if the manual positioning is not accurate enough, the initialization will be failed, and the tracking stage will not start. Second, the map matching can provide distance error as a constraint and eliminate the accumulative error for the long-term navigation. In short, keyframes are aligned with the corresponding SIM and the relative distance is jointly optimized in the FGO as the MD factors.

For each keyframe, an ICP algorithm is implemented to match the corresponding featured points. Note that the mismatching of the ICP will be ignored for the following steps. Given the pose estimation $T_{L,k}^W$ at timestamp k, the transformed feature points transformed in world frame can be represented as,

$$P_k^W = T_{L,k}^W P_k^L \quad (8)$$

where P_k^W and P_k^L denote the keyframes in the LiDAR frame and world frame. Then the error distance of the map matching can be derived as follows:

$$d_k = \sum_{i=1}^{N} D_i(T_{L,k}^{W\,\prime})^2 \quad (9)$$

where $T_{L,k}^{W\,\prime}$ is estimated transformation matrix between the P_k^L and the SIM. D_i denotes the minor operator based on ICP algorithm. Therefore, the error function for the MD constraints can be represented as Equation 10,

$$\|\mathbf{e}_i^{MD}\|^2_{\sum_i MD} = \|d_k - d_0\|^2_{\sum_i MD} \quad (10)$$

Where Σ_{MD} denotes the information matrix of the error function. The error distance is expected to zero according to the MD constraint.

4.4 IMU pre-integration

When raw measurements from IMU are available, the velocity and acceleration can be represented using Equation 11 and Equation 12,

$$\widehat{\omega}_t = \omega_t + \mathrm{b}_t^{\omega} + \mathrm{n}_t^{\omega} \quad (11)$$

$$\hat{\mathrm{a}}_t = \mathrm{R}_t^{\mathrm{BW}}(\mathrm{a}_t - \mathrm{g}) + \mathrm{b}_t^{\mathrm{a}} + \mathrm{n}_t^{\mathrm{a}} \quad (12)$$

where $\widehat{\omega}_t$ and $\hat{\mathrm{a}}_t$ are the raw IMU measurements in the body frame coordinate B, while b and n denote the bias and white noisy, respectively.

Based on the raw data, the relative motion including velocity, position, and rotation based on IMU pre-integration method can be represented as the following Equations (Forster et al., 2017):

$$\begin{aligned}\Delta\mathrm{v}_{mn} &= \mathrm{R}_i^{\mathrm{T}}(\mathrm{v}_n - \mathrm{v}_m - \mathrm{g}\Delta t_{mn})\\ \Delta\mathrm{p}_{mn} &= \mathrm{R}_m^{\mathrm{T}}\left(\mathrm{p}_n - \mathrm{p}_m - \mathrm{v}_m\Delta t_{mn} - \frac{1}{2}\mathrm{g}\Delta t_{mn}^2\right)\\ \Delta\mathrm{R}_{mn} &= \mathrm{R}_m^{\mathrm{T}}\mathrm{R}_n\end{aligned} \quad (13)$$

where $\Delta\mathrm{v}_{mn}$, $\Delta\mathrm{p}_{mn}$ and $\Delta\mathrm{R}_{mn}$ denote the pre-integrated relative pose between the timestamp m and timestamp n.

Based on the relative pose estimated using IMU measurements, the IMU pre-integration factor can be represented as following Equation 14,

$$\|\mathbf{e}_{m\to n}^{IMU}\|^2_{\Sigma_{m,n}} = \left\|f(\mathbf{T}_{B,m}^{W\ -1}\boldsymbol{T}_{B,n}^{W})\right\|^2 \quad (14)$$

Where $\mathbf{e}_{m\to n}^{IMU}$ represents the error function for the relative motion. Σ_{IMU} denotes the information matrix of the IMU factor error function. $\mathbf{T}_{B,m}^W$ and $\boldsymbol{T}_{B,n}^W$ denote the state node at timestamps m and n. f denotes the minimization operation.

5. EXPERIMENTAL RESULTS

In this section, experimental results are presented to evaluate the proposed navigation solution in an opensource SLAM dataset, Urbannav (Hsu et al., 2021). Implementation details and performance analysis are presented as follows.

5.1 Experimental details

To evaluate the performance, all the experiments are implemented on a Linux system with Robot Operating System (ROS) 18.04. As for the data collection in Urbannav dataset, 3D LiDAR (Velodyne 32) and an IMU (Xsens Mti 10) are integrated on the mobile vehicle. OpenStreetMap are downloaded from the website within the trajectory area. The parameters used in this

experiment are given in Table 1. During the evaluation, we compare our proposed solution to two state-of-art algorithms, including LEGO-LOAM and Map aided NDT-based algorithm (Kato et al., 2018; Shan and Englot, 2018). Here, to eliminate the influence of feature association and the sampling rate, we use the global points map as the prior map for the NDT-based navigation.

Table 1. Parameters settings for the experimental evaluation

Parameters	Value	Description
δ_θ	5 m	Keyframe selection
δ_t	15°	Keyframe selection
μ	4	Density
σ	0.2	Down sample rate

5.2 Performance analysis

As shown in Figure 6, the proposed navigation solution can successfully estimate a high accurate trajectory compared to the ground truth trajectory. With the help with the proposed MD constraint, the positioning is still stable when the vehicle moves with a turning (Point A) and after a turning (Point B). In addition, during the long-term straight movement, the accumulative error is eliminated by optimizing the map matching constraints for each keyframes.

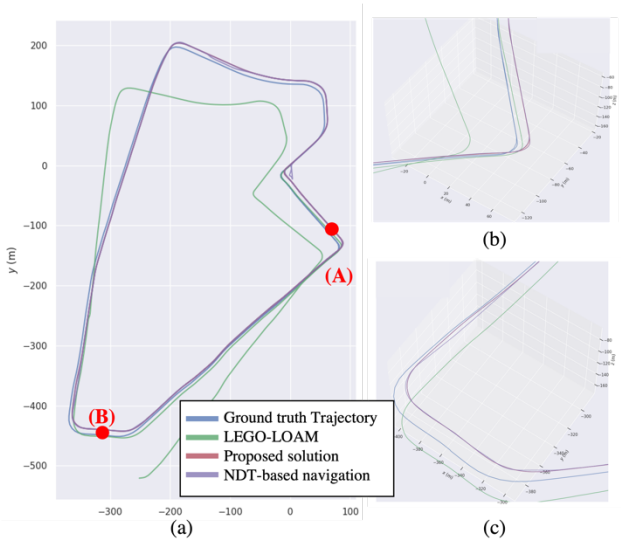

Figure 6. (a) shows the estimated trajectory based on the proposed algorithm, LEGO-LOAM, NDT-based algorithm and ground truth trajectory. (b) and (c) shows the zoom-in comparison at point A and point B.

5.3 Experimental evaluation

Figure 7 shows the trajectory evaluation in translation direction. Compared to LEGO-LOAM solution, the proposed algorithm can significantly improve the accuracy in z value, as shown the location with red arrow. The maximum error from LEGO-LOAM can achieve around 40 meters, while the proposed algorithm can achieve stable positioning error in z direction is deviating from 0 to 3 meters. The reason is that the proposed MD can constraint the accumulating errors even without a global constraint, such as loop closure. In addition, compared to the NDT-based solution, the proposed method involves more measurements and constraints, which achieve better positioning accuracy than it.

To further evaluate the performance of the proposed algorithm, we applied EVO toolkit to calculate the relative error for the translation and rotation using RMSE and Mean (Grupp, 2017). Table 2 and Table 3 show the performance of the translation and rotation of using LEGO-LOAM, NDT-based algorithm, and the proposed algorithm, respectively. Note that we didn't apply the loop closure in the proposed solution to evaluate the contribution of the proposed MD constraints. Regarding the translation, both MEAN and RMSE are reduced in the proposed algorithm. In our experiments, the NDT-based solution also achieves similar accuracy even without the integration of IMU pre-integration. The reason is that both point clouds map and the estimated trajectory are performed on the same datasets, which means that the NDT matching can perform well between the onboard LiDAR and generated point cloud map.

Table 2. Accuracy evaluation in translation (RPE).

Accuracy	Algorithms (m)		
	Lego-LOAM	NDT-based method	Our algorithm
MEAN	19.46	5.72	**4.73**
RMSE	50.00	16.10	**15.26**

Table 3. Accuracy evaluation in rotation (RPE).

Accuracy	Algorithms (degree)		
	Lego-LOAM	NDT-based method	Our algorithm
MEAN	12.87	8.13	**5.36**
RMSE	29.36	15.36	**9.49**

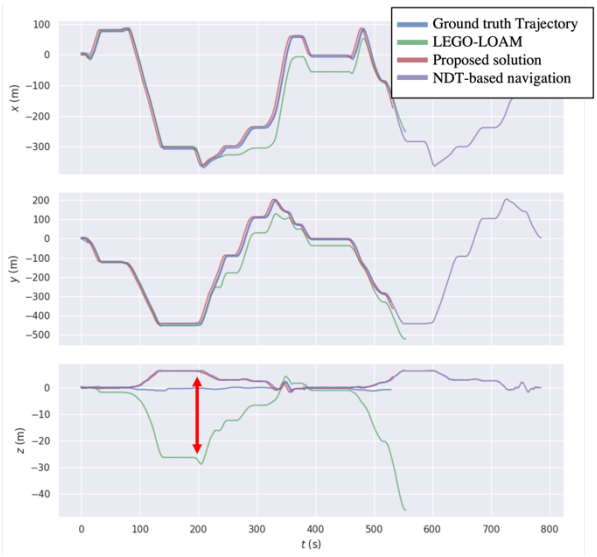

Figure 7. Trajectory evaluation in translation direction.

5.4 Discussion

To evaluate the improvement of the proposed MD, we compare our proposed solution with the pure LiDAR-Inertial odometry (LIO) estimation solution. The pure LIO solution is implemented based on the integration of LiDAR and 6 axis IMU, with the loop closure constraints (Shan et al., 2020). The absolute errors in three translation direction are shown in Table 4. Compared to the LIO solution, the mean errors in x direction and y direction are significantly reduced, with the help of the proposed MD constraints. The error in z direction is similar, with the pre-integration of IMU.

Table 4. Accuracy evaluation in translation

Algorithm	Accuracy (Absolute Error, m)		
	MEAN (X)	MEAN (Y)	MEAN (Z)
LIO solution	33.92	56.83	1.61
Our algorithm	**4.41**	**4.12**	2.21

6. CONCLUSION AND FUTURE WORK

In this work, we propose a navigation solution that combines a LiDAR-Inertial system with OpenStreetMap to enhance the positioning performance in urban canyon environments. We introduce a distance constraint based on map matching and integrate this constraint with LiDAR odometry and IMU pre-integration within a factor graph optimization structure. The evaluation results demonstrate that the proposed solution can achieve high accuracy in urban canyon environments.

One of the limitations of the work is that manual corrections for the consistent evaluation are necessary before the navigation task since the OpenStreetMap may open-source dataset that may not always be up-to-date and are less height information. Deviation of the OSM or wrong elements will decrease the accuracy of map matching and bring in wrong constraints for the whole navigation system. In our future work, we will study the automatically updating for the OSM and the generated SIM to enable large-scale navigation.

ACKNOWLEDGEMENTS

This research was supported by funding provided to Professor Naser El-Sheimy from NSERC CREATE and Canada Research Chairs programs. In addition, we would like to acknowledge the OpenStreetMap and OpenStreetMap Building for the all the datasets used in this paper.

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

Bender, P., Ziegler, J., Stiller, C., 2014. Lanelets: Efficient map representation for autonomous driving. IEEE Intell. Veh. Symp. Proc. 420–425. https://doi.org/10.1109/IVS.2014.6856487

Chen, X., Milioto, A., Palazzolo, E., Giguere, P., Behley, J., Stachniss, C., 2019. SuMa++: Efficient LiDAR-based Semantic SLAM. IEEE Int. Conf. Intell. Robot. Syst. 4530–4537. https://doi.org/10.1109/IROS40897.2019.8967704

Chiang, K., Chiu, Y., Srinara, S., Tsai, M., 2023. Performance of LiDAR-SLAM-based PNT with initial poses based on NDT scan matching algorithm. Satell. Navig. 4. https://doi.org/10.1186/s43020-022-00092-0

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

Elhousni, M., Zhang, Z., Huang, X., 2022. LiDAR-OSM-Based Vehicle Localization in GPS-Denied.

Forster, C., Carlone, L., Dellaert, F., Scaramuzza, D., 2017. On-Manifold Preintegration for Real-Time. IEEE Trans. Robot. 33, 1–21.

Frosi, M., Gobbi, V., Matteucci, M., 2023. OSM-SLAM: Aiding SLAM with OpenStreetMaps priors. Front. Robot. AI 10, 1–12. https://doi.org/10.3389/frobt.2023.1064934

Grupp, M. 2017. evo: Python package for the evaluation of odometry and SLAM. https://github/com/MichaelGrupp/evo

Gui, L., Zeng, C., Luo, J., Feng, H., 2022. Prior Map Aided LiDAR-Based Localization Framework via Factor Graph. CEUR Workshop Proc. 3248.

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.

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., Monrroy, 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

Li, Jing, Qin, H., Wang, J., Li, Jiehao, 2022. OpenStreetMap-Based Autonomous Navigation for the Four Wheel-Legged Robot Via 3D-Lidar and CCD Camera. IEEE Trans. Ind. Electron. 69, 2708–2717. https://doi.org/10.1109/TIE.2021.3070508

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, T., Chang, L., Niu, X., Liu, J., 2020. Pole-like object extraction and pole-aided gnss/imu/lidar-slam system in urban area. Sensors (Switzerland) 20, 1–19. https://doi.org/10.3390/s20247145

Pai, H.-Y., Zeng, J.-C., Tsai, M.-L., Cheng, K.-W., 2022. Automated Modelling of Road for High-Definition Maps With Opendrive Format Utilizing Mobile Mapping Measurements. Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci. XLIII-B1-2, 263–270. https://doi.org/10.5194/isprs-archives-xliii-b1-2022-263-2022

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

Yan, F., Vysotska, O., Stachniss, C., 2019. Global Localization on OpenStreetMap Using 4-bit Semantic Descriptors. 2019 Eur. Conf. Mob. Robot. ECMR 2019 - Proc. https://doi.org/10.1109/ECMR.2019.8870918

Zhang, J., 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