Journal of Geodesy (2023) 97:39
https://doi.org/10.1007/s00190-023-01728-y

ORIGINAL ARTICLE

A ubiquitous positioning solution of integrating GNSS with LiDAR odometry and 3D map for autonomous driving in urban environments

Jingbin Liu[1] · Yifan Liang[1] · Dong Xu[1] · Xiaodong Gong[1] · Juha Hyyppä[2]

Received: 19 July 2022 / Accepted: 16 March 2023

Abstract

Continuous and reliable positioning anywhere anytime is one of the critical technologies for autonomous driving, whereas it is challenging yet, especially in urban environments. Due to the presence of buildings, Global Navigation Satellite System (GNSS) is not reliable for degraded positioning accuracy and availability. With the simultaneous localization and mapping (SLAM) approach, Light detection and ranging (LiDAR) has been widely used by autonomous driving vehicles for localization and environmental perception. However, LiDAR odometry (LO) or SLAM is not accurate enough for autonomous driving as it suffers from accumulative errors. This study proposes a GNSS/LiDAR/Map integrated positioning solution to provide ubiquitous positioning for autonomous driving, especially in urban areas, including urban canyons and underground, where GNSS is blocked partly or even fully. Matching LiDAR scans with a 3D map, named 3D map-based global localization (MGL), can provide an absolute positioning solution with respect to the map. A drift error model of LiDAR odometry is proposed to improve the accuracy of LiDAR odometry, and its model parameters are estimated online using either GNSS or MGL absolute locations. Using the graph optimization approach, GNSS and/or MGL precise absolute positioning are integrated with LO relative positioning to provide continuous and reliable positioning even when GNSS is degraded. The proposed solution is validated using the NCLT public dataset, which contains LiDAR data for building 3D maps and 3D map-based global localization, respectively. Based on the NCLT dataset, we simulate three typical urban scenarios to verify the performance of ubiquitous positioning. The experiments show that the proposed drift error correction method decreases accumulated error of LiDAR odometry by 35.5%, and the proposed GNSS/LiDAR/Map solution can provide continuous positioning with the availability of 100%, and improves the positioning accuracy by 49.8%, compared to the state-of-the-art approach of GNSS/LO fusion.

Keywords Indoor positioning · Underground positioning · Global localization · Integrated positioning · Multi-sensor positioning

1 Introduction

Autonomous driving depends on continuous and precise positioning anywhere anytime during its whole course for navigation and environmental perception ((Joerger and Spenko, 2017; Stephenson et al., 2011)). Using the RTK (Real-time kinematic) or PPP (Precise Point Positioning) technologies, Global Navigation Satellite System (GNSS) can provide precise positioning in open-sky conditions, while the accuracy and availability of GNSS positioning are not satisfactory in urban environments, e.g., urban canyons, underground parking and tunnels, due to the signal block-

✉ Yifan Liang
lyf0312@whu.edu.cn

Jingbin Liu
jingbin.liu@whu.edu.cn

Dong Xu
dongxu@whu.edu.cn

Xiaodong Gong
gongxiaodong@whu.edu.cn

Juha Hyyppä
juha.hyyppa@nls.fi

[1] State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing, Wuhan University, Wuhan, China

[2] Department of Remote Sensing and Photogrammetry, Finnish Geospatial Research Institute, Espoo, Finland

Published online: 25 April 2023

age (Julier and Durrant-Whyte 2003). As Light detection and ranging (LiDAR) is widely equipped with autonomous driving vehicles, the state-of-the-art approach integrates GNSS and LiDAR odometry to provide continuous positioning. However, as LiDAR odometry (LO) is relative positioning, and it suffers from accumulative errors, the accuracy of current GNSS/LO fusion approach depends on the accuracy and availability of GNSS precise positioning. In urban environments where GNSS is not available for a long time, the accuracy of GNSS/LO fusion positioning is not adequate for autonomous driving. Therefore, continuous and precise positioning in urban environments is challenging yet (Li et al. 2019).

LiDAR has been used in autonomous driving for localization and environmental perception (Lu et al. (2020); Yurtsever et al. (2020)). LiDAR-based simultaneous localization and mapping (LiDAR-SLAM) has been widely discussed and improved over the last decade (Wang et al. 2019). By using perception sensors (LiDAR) to derive the navigation information, the SLAM algorithm is able to simultaneously localize itself and map the environment. A SLAM system usually includes front-end odometry and back-end optimization (Hess et al. 2016). The front-end calculates the relative pose of each LiDAR scans by point cloud registration (Ma et al. 2022), also called LiDAR odometry, and the back-end optimizes the position by fusing other information. To reduce the point cloud information and represent the environment clearly, grid-based SLAM transforms the 2D point cloud into an occupancy grid map. Gmapping is one of the most widely used grid-based SLAM techniques for mobile robots, utilizing Rao-Blackwellized particle filters (RBPFs) to achieve real-time positioning and mapping (Grisetti et al. 2007). The usage and the core scan matching approach used in Hector SLAM are more flexible and adaptable (Kohlbrecher et al. 2011). The main drawback of this algorithm is that it is heavily dependent on the initial position and heading. The traditional iterative closest point (ICP) is well known as the 6DoF LiDAR-SLAM registration method. It adopts the point-to-point matching LiDAR-SLAM as well as the point-to-plane and plane-to-plane methods (Holz et al. 2015). Instead of considering the whole point cloud, normal distribution transform (NDT) uses surface information to calculate the likelihood modeled by a linear combination of normal distributions (Magnusson 2009). However, traditional NDT matching methods do not consider the weight of different types of surfaces, an improved NDT matching method is proposed in our study. LiDAR odometry and mapping (LOAM) are presently considered as a state-of-the-art 6DoF SLAM approach especially when new modifications, such as A-LOAM, Lego-LOAM, F-LOAM, and LIO-SAM are taken into account (Ji and Singh 2014; Shan and Englot 2019; Shan et al. 2020; Wang et al. 2021; Zhang and Singh 2017). It employs a feature-based registration method to extract the edges and planar features in a consecutive point cloud. In the work Ji and Singh (2014), the outdoor tests have a relative accuracy of roughly 2.5%. Since there are still errors in each relative position between scans, and these errors will gradually accumulate, the positioning accuracy of SLAM will decrease over the distance. On the other hand, the SLAM-derived solutions are relative positions in the sensor coordinate system, which makes SLAM unable to be used alone.

Different to the SLAM approach, which can only provide relative position and pose, the global localization approach can provide absolute localization solutions by matching LiDAR scans with a previously built 3D map, which is called in this study 3D map-based global localization (MGL). In this paper, the 3D map refers to 3D point cloud map, not the products generated by 3D point cloud. The matching between LiDAR scans and a 3D map is based on the similarity of variety of descriptors, which can be defined either manually or by using deep learning. The key to global localization is whether the descriptor can describe correctly the feature of a LiDAR scan. Bosse and Zlot (2013) describes scans using 3D Gestalt descriptors directly and matched LiDAR scans with a map using a vote-matrix representation. Some approach converts a LiDAR scan to a range-image and calculates feature descriptors by using image feature such as Speeded Up Robust Features (SURFs) (Rusu et al 2009; Steder et al. 2010). Different to the method using simple geometry feature, Finman et al. (2015) proposed an object-level method that detect objects and used all object descriptors to represent a point cloud, but this method only works for RGB-D cameras and only a small number of objects was used. The deep learning-based method opens a completely new window to solving the global positioning problem in a data-driven fashion. Zeng et al. (2017) proposed the 3DMatch method, one data-driven 3D convolutional neural network (CNN) for learning geometry features from RGB-D reconstruction data. Elbaz et al. (2017) proposed an algorithm that extracts the geometry feature of a point cloud by deep neural network auto-encoder, and they improved the selection of the basic units for matching to reduce the computational complexity. L3-Net is a learning-based LiDAR localization network established by various deep neural networks such as Recurrent Neural Networks (RNNs) and 3D convolutional neural networks (3D CNNs) (Lu et al. 2020). The end-to-end neural networks-based method is an emerging approach, which directly takes the LiDAR scans as input and the global position as output. DeepICP is the first end-to-end point cloud registration framework. In addition to extracting descriptors through neural networks, this method could directly match the local and target point clouds instead of using ICP algorithms (Lu et al. 2019). Given the coordinate reference of a 3D map, LiDAR-based global localization provides an absolute positioning solution with the accuracy of roughly 10 cm

(Xu et al. 2022b). However, this method depends on the availability of 3D map data, and it cannot output high-frequency positioning solutions due to the high computational load.

Many studies have reported the integration of SLAM and GNSS, including tightly coupled and loosely coupled approaches, to achieve better performance in terms of positioning accuracy and continuity. Li et al (2020b); Qian et al. (2020) integrated SLAM with GNSS to enhance ambiguity resolution and improve GNSS positioning accuracy in GNSS partly denied environment. In Qian's work, a LiDAR-based fuzzy one-to-many feature matching method is reported to aid RTK single-frequency single-epoch ambiguity resolution, which improved positioning accuracy and fixed rate of RTK in urban environments.

The GNSS/SLAM loosely coupled approach uses either filtering-based or optimization-based algorithms to reduce the cumulative error of SLAM (He et al. 2021). Extended Kalman Filter (EKF) and Rao-Blackwellized particle filter (RBPF) have been commonly used for fusing GNSS, INS, and LiDAR-SLAM to achieve 6DOF positioning (Gao et al. 2015; Kim and Sukkarieh 2005; Schultz et al. 2016; Shamsudin et al. 2018). In recent years, the graph optimization-based fusion approach has received more and more attention, as it can fuse flexibly different types of positioning sensors, such as GNSS, wheel odometer, LiDAR, camera, and IMU (Bhamidipati and Gao 2020; Chang et al. 2019; Kukko et al. 2017; Pierzchała et al. 2018). Chang et al. (2019) incorporates GNSS and IMU pre-integration results into the SLAM back-end optimization, and uses a sliding window to ensure that the computational load of back-end does not increase with time. In Pierzchała et al. (2018)'s work, four sensors are fused using graph optimization, including multi-line LiDAR, stereo camera, IMU, and GNSS. With the post-processing, the method achieves good 3D mapping performance for forest surveying, while the real-time positioning performance is not adequate. GNSS/SLAM loosely coupled approach requires at least a limited availability of GNSS precise positioning, and the fusion solution has unsatisfactory performance when GNSS is not available for a long time. In general, compared to the optimization-based approach, the filtering-based approach has higher computational efficiency, but lower positioning accuracy (Chang et al. 2019).

In this paper, we propose a ubiquitous positioning solution by integrating LiDAR odometry, GNSS, and 3D Map matching-based global localization using the graph optimization (GO) algorithm to provide continuous and precise positioning for autonomous driving vehicles, especially in urban environments. 3D map-based global localization solution has the same geo-reference system as the 3D map. As 3D map data is typically geo-referred using a GNSS-based high-precision positioning system in the mapping data collection process (Novatel 2022), 3D map-based global localization solution has the same coordinate system as GNSS positioning. Given the wide availability of 3D maps, the proposed GNSS/LiDAR/map integrated positioning solution can provide continuous and precise positioning, and a drift error correction model of LO is proposed to further improve the positioning accuracy when neither GNSS nor MGL works. To summarize, our main contributions include:

This study proposes a ubiquitous positioning solution by integrating relative positioning, e.g., LiDAR odometry, and absolute positioning, e.g., GNSS and MGL, for continuous and precise positioning of autonomous driving in complex scenarios of urban environments, e.g., from open-sky, GNSS partly blocked to fully denied environments. The proposed solution exploits the graph optimization method to fuse all available measurements adaptively according to their uncertainties.

An online estimated drift error correction model is incorporated with the proposed integrated solution to correct the drift error of LiDAR odometry. The model parameters are estimated using absolute positioning solutions, e.g., either GNSS or 3D map matching-based global localization. Then the model is applied to improve the accuracy of ubiquitous positioning, especially when absolute positioning solutions, e.g., either GNSS or MGL, are not available.

Three typical scenarios are set up to simulate common urban scenes with different availabilities of GNSS and MGL, e.g., in underground parking lots and urban canyons. The proposed solution is validated by evaluating the positioning accuracy and availability in these three scenarios.

The remainder of this paper is organized as follows. The methodology of the proposed solution is described in Sect. 2. Experimental implementation and the results are presented in Sect. 3. Section 4 provides the discussions and conclusions.

2 Methodology

2.1 System overview

This section describes the architecture of the proposed GNSS/LiDAR/map integrated positioning solution as shown in Fig. 1. The proposed solution integrates three components: LiDAR odometry of relative positioning, GNSS precise positioning, and 3D map matching-based global localization of absolute positioning. Given their uncertainty levels, these relative and absolute positioning solutions are integrated using the graph optimization algorithm. Within the architecture, the drift error correction model of relative positioning is estimated online and is used for improving the accuracy of the proposed integrated positioning solution, as shown in Fig. 1.

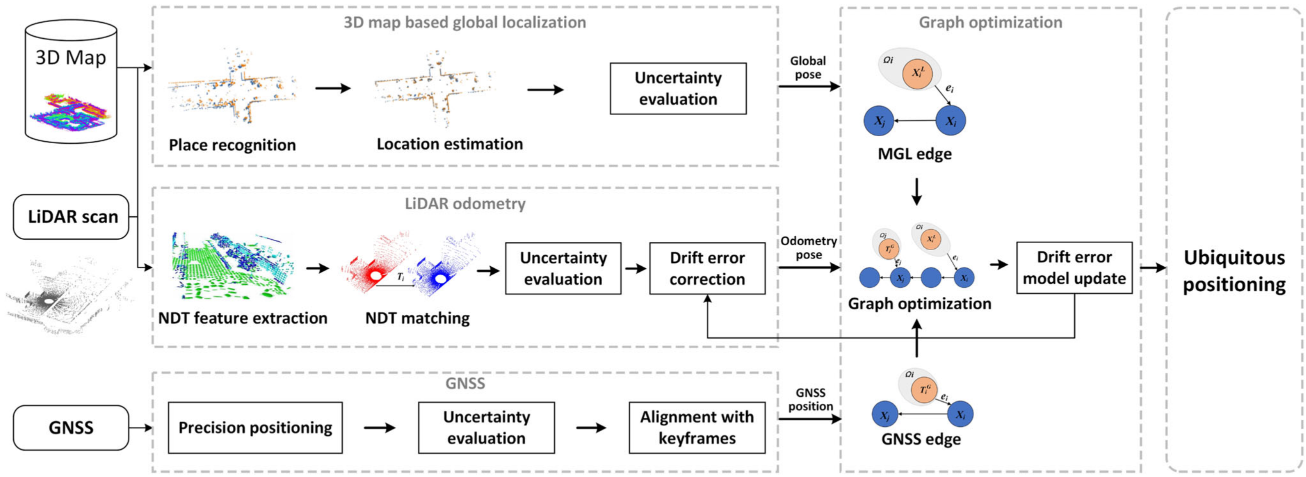


Fig. 1 Overview of proposed ubiquitous positioning solution

2.2 3D map matching-based global localization (MGL)

3D map is one of the infrastructures for autonomous driving, and it has been widely available. We use the 3D map data to provide an absolute positioning solution using the one-shot LiDAR global localization method, and the absolute positioning solution is integrated into our proposed solution. The one-shot LiDAR global localization method has been presented in the work (Xu et al. 2022a), and we reiterate simply the method for completeness.

The one-shot LiDAR global localization method matches one LiDAR scan with the pre-defined 3D map to resolve the 6DOF position and pose of the scanner with respect to the 3D map. The global localization method consists of two steps: place recognition and pose estimation. As presented in Xu et al. (2022a), the place recognition method defines the polar grid height coding image (PGHCI) descriptor, so that the method is featured with high descriptiveness and rotational-invariance. More importantly, by introducing the PGHCI descriptor and the concept of virtual LiDAR, the proposed place recognition method in Xu et al. (2022a) breaks through the limitation of previous methods that require the same types of LiDAR scanners in the mapping process and localization process.

Given the place recognition provides a coarse location, Xu et al. (2022a) presented a one-shot global localization solution that matches only one single LiDAR scan with 3D map data to resolve the 6DOF pose. The one-shot global localization solution consists of a high-quality CSSC descriptor and a novel pipeline method, including two-stage similarity estimation, Nearest Cluster Distance Ratio (NCDR), and the Selective Generalized Iterative Closest Point (SGICP) algorithm. The proposed CSSC descriptor features an enhanced high descriptiveness using the elevation weight and point density weight. As a result, the one-shot global localization method achieves the positioning accuracy of 7 cm-18 cm for different datasets with different types of LiDAR scanners in the mapping process (Xu et al. 2022b).

Furthermore, the work (Xu et al. 2022b) develops an integrity index to evaluate the integrity and precision of the presented solution. Correspondingly, the integrity index is used in graph optimization to determine the information matrix of the MGL solution. When the integrity index is zero, the solution of MGL will not be used in graph optimization. Otherwise, the information matrix of the MGL solution will be calculated using the integrity index.

2.3 An improved LiDAR odometry and drift error correction model

In this section, we introduce an improved LiDAR odometry solution based on NDT matching algorithm, which is one of three components in the proposed integrated positioning solution. Furthermore, a drift error correction model is proposed to correct the LiDAR odometry.

2.3.1 Weighted NDT-based LiDAR odometry

The NDT algorithm has been reported in Magnusson (2009), and it is efficient for registration. For 3D LiDAR, NDT shows a better performance than other scan matching algorithms, such as Iterative Closest Points, in terms of both reliability and processing speed (Magnusson 2009). However, the original NDT algorithm constructs the matching target equation using the same weight for all voxels. In fact, the distance between the center of the voxel and the scanning center, as well as the surface characteristics of the voxel will affect the calculation of transformation parameters. This study applies a weighted NDT algorithm in the LiDAR odometry, which has been validated by Chen (2020).

Firstly, the relation between scan distance and weight is discussed. The same rotation error results in greater inconsistencies in voxels farther from the scanning center than in voxels closer to the scanning center. In other words, the greater the distance between the center of the voxel and the scanning center, the stronger the constraint of the voxel on the matching. Therefore, distance weights are introduced:

$$W_{r_k} = |x_k| \tag{1}$$

where x_k is the center of voxel k and W_{r_k} is the distance weight of voxel k.

Second, the NDT algorithm describes the surface properties of local point clouds in a concise way. In this method, the local point cloud is projected into a series of smooth surfaces composed of local PDFs (Probability Density Function). Each PDF describes the shape and surface characteristics of the local surface in its corresponding voxel by means of mean and variance. The covariance Σ of each voxel is a positive definite matrix, and the eigenvalues and eigenvectors of the matrix can be obtained by eigenvalue decomposition. The three eigenvalues of the covariance matrix Σ are all positive and are arranged in descending order as: $\lambda_1 \geq \lambda_2 \geq \lambda_3 > 0$. According to the eigenvalues and eigenvectors of the covariance matrix sigma, a variety of geometric parameters of the voxel can be calculated. West et al. (2004) introduces the calculation methods of several parameter indexes. The following indexes are used to express linearity index, planarity index, and sphericity index in the voxel:

$$\begin{gathered} \forall j \in [1,3],\ \sigma_j = \sqrt{\lambda_j} \\ a_{1D} = \frac{\sigma_1 - \sigma_2}{\sigma_1},\ a_{2D} = \frac{\sigma_2 - \sigma_3}{\sigma_1},\ a_{3D} = \frac{\sigma_3}{\sigma_1} \end{gathered} \tag{2}$$

The dimensional of the voxel (1D, 2D or 3D) is defined as follows:

$$d^* = \underset{d \in [1,3]}{\operatorname{argmax}} \left[a_{dD}\right] \tag{3}$$

Voxels with different dimensional indices have different influences on NDT scan matching. The closer the surface inside the voxel is to the 2D plane, the more accurately the local PDF can describe the real physical surface of the point cloud. Therefore, voxels with a higher plane index or significant 2D characteristics contribute more to NDT matching and should be given a higher weight. Due to the sparsity of the point cloud, the linearity or 1D features do not represent accurately real physical surfaces. The LiDAR we used is a rotating LiDAR, it includes multiple scanning lines. The farther away from the scanning center, the lower the point cloud density, as Fig. 2, a plane maybe just a few line segments in the sparse point cloud. In a voxel far from the scanning center, it may contain a line, but it might just be part of a plane. The linearity or 1D features should be assigned lower weights.

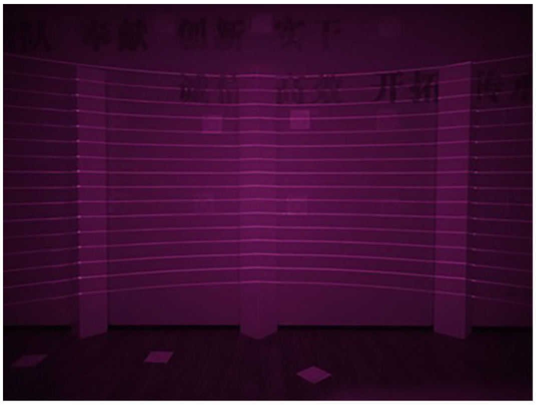


Fig. 2 Rotating LiDAR scan is sparse in far away. A whole wall far from the scanning center is a few line segments in the sparse point cloud

In this paper, the weights are set as follows:

$$WD = \begin{cases} 1.25 & (d^* = 2) \\ 1.0 & (d^* = 3) \\ 0.75 & (d^* = 1) \end{cases} \tag{4}$$

Therefore, the objective equation of weighted NDT scan matching is:

$$\begin{aligned} F(\xi) &= \sum_{k=1}^{n} W_k \tilde{f}\left(T\left(\xi, q_k\right)\right) \\ &= \sum_{k=1}^{n} -d_1 W_k \cdot \exp\left(-\frac{d_2}{2}\left(T\left(\xi, q_k\right) - \mu_k\right)^T \Sigma^{-1}\left(T\left(\xi, q_k\right) - \mu_k\right)\right) \\ T\left(\xi, q_k\right) &= \exp\left(\xi^\wedge\right) \cdot q_k = R q_k + t, \\ W_k &= W r_k \cdot W D_k \end{aligned} \tag{5}$$

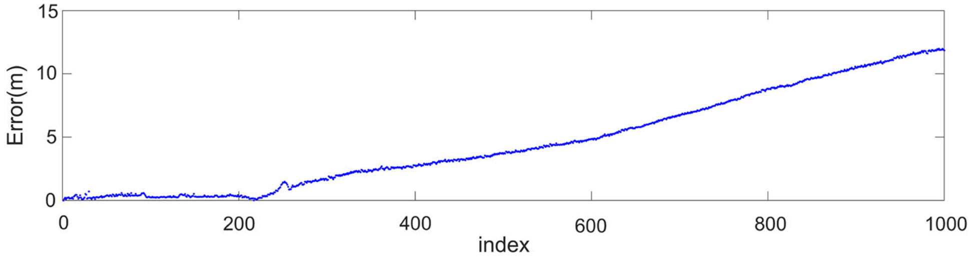
After getting the transformation T between the two LiDAR scans, current position relative to the first LiDAR scan is obtained by accumulating all these transformations.

2.3.2 Drift error correction model

When the LiDAR odometry calculates the relative trajectory by matching consecutive LiDAR scans recursively, the registration error between scans will be accumulated, and the LiDAR odometry has accumulative error over time, as shown in Fig. 3. When absolute positioning solutions are not available for a long time, the integrated positioning solution depends on the LO for a full availability, while its accuracy is to be improved yet.

This study proposes a drift error correction (DEC) model to improve positioning accuracy. As shown in Fig. 4, suppose

Fig. 3 Errors of LiDAR odometry accumulate over time



the current time is i, there are absolute positioning solutions (either GNSS or MGL) available at time i, and a previous time j, while there are no absolute positioning solutions after time i. Time j is separated from the current time i by a certain time interval, such as 10 s. We assume the pattern of error accumulation is the same in short periods, as the error of LO is derived from point cloud registration. The DEC model is estimated using the reference trajectory between time i and j, which is derived with the absolute positioning solutions between time i and j. The estimated DEC model is then used for correcting the LO trajectory of the following period after time i, e.g., the time k in Fig. 4. Behind this methodology, it is assumed that the accumulated error of the LO behaves in part with the nature of systematic error when the scanning environment has no significant change, and thus the LO error can be modeled and predicted in a short period (Ji and Singh 2014; Li et al. 2020a).

We use the trajectory estimated with the graph optimization as the reference trajectory, and the error is defined as the difference between the real-time LO trajectory and the reference trajectory. The drift error correction model is defined as Eq 6.

$$\Delta e = \frac{\left(p_i^l - p_i^g\right)}{\sum_{n=j}^{i-1} \left\| p_{n+1}^g - p_n^g \right\|} \tag{6}$$

where p_i^l and p_i^g are the position of time i from LiDAR odometry and graph optimization, respectively.

When an absolute positioning solution (either GNSS or MGL) is available, the DEC model can be updated using Eq 6. Subsequently, when there are no absolute positioning solutions, the LO is the main observables of the integrated solution. The updated DEC model is then used to correct the LO, and improve the accuracy of the integrated positioning solution. As shown in Fig. 4, given the time k and the corresponding LO position p_k, the corrected position is calculated using Eq. 7.

$$p_k' = p_k + \Delta e \sum_{n=i}^{k-1} \left\| p_{n+1} - p_n \right\| \tag{7}$$

where p_k is the position of time k from LiDAR odometry, p_k' is the position of time k after drift error correction, and Δe is the error model of LiDAR odometry which be updated by Eq 6.

Finally, the corrected position and pose p_k' will be output in real-time as the integrated positioning solution. It has an improved accuracy than the original LO, and can also provide a better initial value for graph optimization.

2.4 GNSS/LiDAR/Map fusion using Graph optimization

This section presents first the fusion framework of integrating the LO, GNSS, and MGL using the graph optimization algorithm, and then describes the construction of the optimization equations.

The proposed fusion framework integrates the LO of relative positioning, and GNSS and MGL of absolute positioning using the graph optimization algorithm (Grisetti et al. 2010), as shown in Fig. 5. The fusion framework exploits the advantages of these different technologies and mitigates the drawbacks of their own. GNSS and MGL are able to provide previse absolute positioning with the accuracy of up to a number of centimeters, while their continuity, availability, and accuracy are affected by the surrounding environments. LO technology can provide continuous relative positioning with a high rate, e.g., 10Hz (Ji and Singh 2014), while it suffers from increasingly accumulative drift error over time, and the accuracy is not satisfactory.

As shown in Fig. 5, the states (X), (i.e., positions of autonomous driving vehicles), are the nodes in the graph optimization framework, and the transformation between two consecutive states is defined as motion edge. As to the LO, a series of key frames are constructed based on movement distance, the relative pose of the current key frame is associated in the graph as another measurement edge. GNSS and MGL absolute positioning solutions are firstly aligned temporally and spatially, and they are associated in the graph as the measurement edges. It should note that for each node, the key frame of LO is available, while GNSS and MGL measurement edges are not necessarily available.

Given the methodology of graph optimization, for the LO, let $\hat{z}_{ij}\left(x_i, x_j\right)$ be the prediction of a virtual measurement given a configuration of the nodes x_i and x_j. Usually, this prediction is the relative transformation between the two nodes.

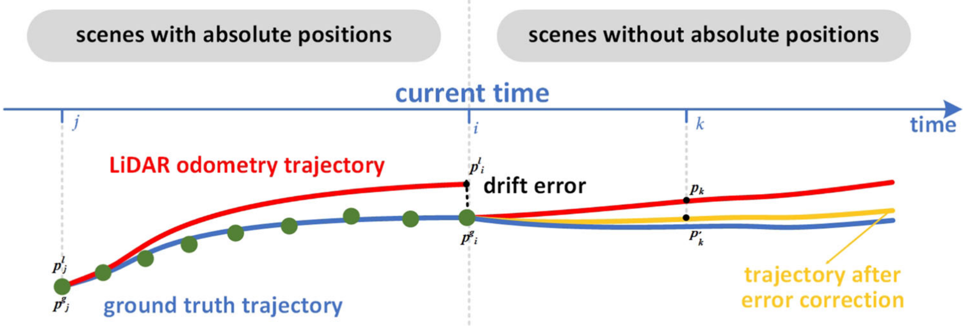

scenes with absolute positions
scenes without absolute positions
current time
j
i
k
time
LiDAR odometry trajectory
p^l_i
drift error
p_k
p'_k
p^g_i
p^l_j
p^g_j
ground truth trajectory
trajectory after error correction

Fig. 4 Illustration of the accumulative error of LO and the DEC model. The red lines are the trajectory of LiDAR odometry that contains increasingly accumulative error over time, and the yellow line is the corrected LO trajectory using the DEC model. The blue line is the ground truth trajectory used as the reference. Green points are the positions that are estimated using the graph optimization of the proposed solution (section 2.4). The blue axis represents time, and the current time is *i*

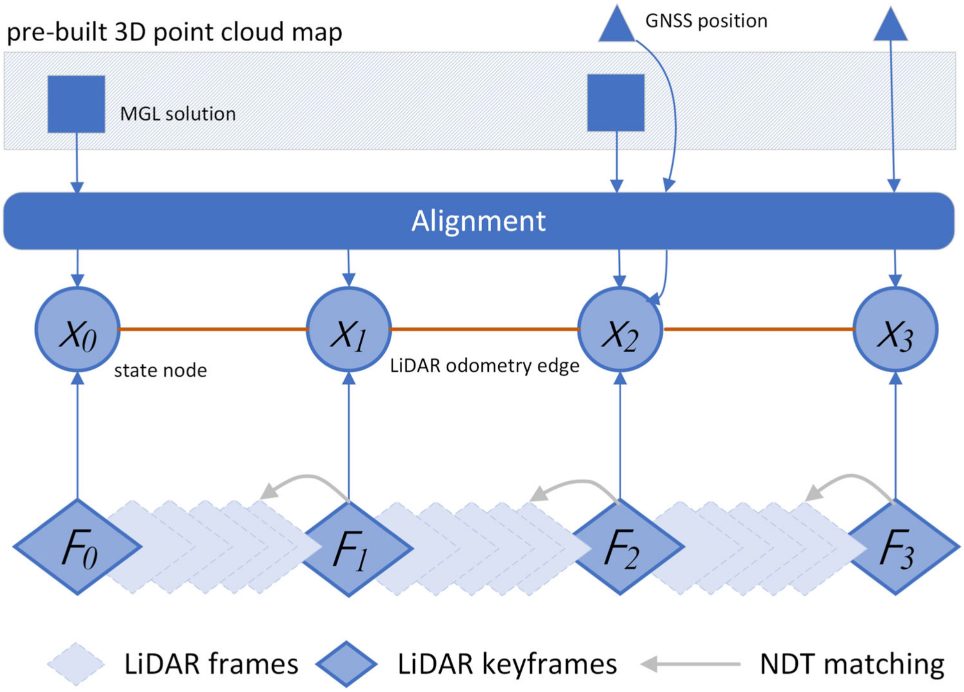


Fig. 5 Structure of graph optimization

The error function between the expected observation $\hat{z}_{ij}$ and the real observation z_{ij} is simply given by:

$$e_{ij}\left(x_i, x_j\right) = z_{ij} - \hat{z}_{ij}\left(x_i, x_j\right) \tag{8}$$

The graph optimization accounts for the uncertainty of the measurements using the information matrix Ω_{ij}. For the LO, the information matrix of this edge is calculated based on the fitness score of two consecutive key frames as ¨ij $= score \circ I$, where *score* is the fitness score and I is the identity matrix.

For GNSS absolute positioning solution, the error function is defined as the difference between GNSS-derived position and the expected position of the current node: $e_G = \hat{z}_i - z_G$.

The information matrix of GNSS positioning solution is calculated using the corresponding covariance matrix and the solution state (a fixed or float solution) as

$$\Omega_G = \begin{cases} score_{fix} \circ \boldsymbol{I}, & RTK_{state} = \text{FIX} \\ score_{float} \circ \boldsymbol{I}, & RTK_{state} = \text{FLOAT} \\ 0, & others \end{cases},$$

where $score_{fix}$ and $score_{float}$ are the corresponding weight parameters for RTK fix and float solutions, respectively. Furthermore, the covariance matrix in the RTK positioning results can also be used to accurately calculate these two parameters.

For the MGL absolute pose, the error function is defined as the difference between MGL-derived pose and the expected pose of the current node as $e_L = \hat{z}_i - z_L$.

The information matrix of MGL pose is calculated using the integrity score as $\Omega_L = WCS \circ \boldsymbol{I}$, where WCS is the weighted combined score of integrity index of the MGL solution, and its calculation is presented in Xu et al. (2022b).

Given the node and edges, the optimization function is defined as follows:

$$\begin{aligned} F(x) &= \sum_{k=1}^{n} e\left(x_k, z_k\right)^T \Omega_k e\left(x_k, z_k\right) \\ \hat{x} &= \arg\min_x F(x) \end{aligned} \tag{9}$$

where $x = \left(x_1^T, \ldots, x_n^T\right)^T$ is a vector of parameters to be estimated, where x_i describes the pose of node i, and each x_k describes the state of node k, which contains position and pose. z_k and Ω_k represent, respectively, the mean and the information matrix of a constraint relating node and the constraint. e is a vector error function that measures how well the parameter blocks x_k satisfy the constraint z_k.

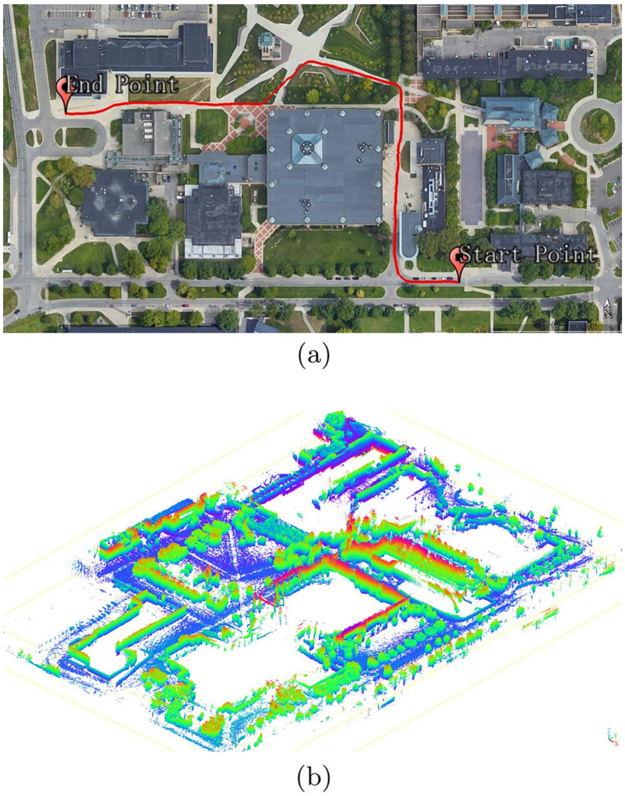
According to the above equation, graph optimization can be seen as a non-linear least squares problem essentially, and it can be solved by methods such as Gauss–Newton or Levenberg–Marquardt (L–M) method.

The result of graph optimization is the optimized state of every node, which represents the optimized positions of an autonomous driving vehicle. The optimized positions are used to update the drift error correction model and provide ubiquitous positioning for an autonomous driving vehicle.

3 Experiments

3.1 Experiment setup and ground truth

The proposed integrated positioning solution is implemented using C/C++ programming language, and it is validated using the testing experiments and the NCLT open-source dataset(Carlevaris-Bianco et al. 2016). In order to validate the proposed methods, the experiment requires a dataset containing GNSS precise positioning, and two sets of LiDAR scanning measurements of the same area, one of which is used for building a 3D map, and the other is used as global localization with the MGL method. Common SLAM open datasets contain only one LiDAR scanning data, so we use the open-source dataset NCLT as the experimental data in this study, which contains multiple sets of LiDAR scanning measurements in the same area. As shown in Fig. 6a, the NCLT dataset was acquired at the campus of the University of Michigan in USA using the set of the sensors, including a Velodyne HDL-32E LiDAR, GNSS RTK, IMU, wheel encoders, among others. As the MGL method requires a 3D map as the geospatial reference, we build first a 3D point cloud of the scene as shown in Fig. 6b. (The reference map consists of each LiDAR scan and its 6DOF pose in a global reference frame. The 6DOF pose of LiDAR scans is provided by the ground truth of the dataset and transformed from the robot's origin to LiDAR origin.) The ground truth of the trajectories is provided using the combined GNSS/LiDAR with the post-processing. The accuracy evaluation of the MGL method with the NCLT open-source dataset is referred to our previous work (Xu et al. 2022b).



(a)

(b)

Fig. 6 **a** The segmented test route; **b** The corresponding 3D point cloud map of the testing area

We use the absolute trajectory error (ATE) to evaluate the localization accuracy. The absolute trajectory error directly measures the difference between points of the true and the estimated trajectory. As a pre-processing step, we associate the estimated positions with ground truth positions using the timestamps. Then, we compute the difference between each pair of positions as the errors, and calculate the mean/median/standard deviation of these differences (errors). The 'Mean' is short for Mean Absolute Error, the 'RMSE' is short for Root Mean Square Error, the 'Max' represents the max error, and the 'Std' is short for Standard Deviation.

3.2 Validation of DEC model

In this subsection, we will compare the positioning accuracy of LiDAR odometry with and without the proposed DEC model. In this experiment, there are two trajectories to compare. One trajectory is the integrated NDT-based LiDAR odometry (mentioned in section 2.2) with GNSS and MGL solutions directly. When an absolute position from GNSS or MGL is available, the current position of LiDAR odometry will be replaced with this absolute position and subsequent positioning results will be derived from this absolute position. Another trajectory is improved using the proposed DEC model based on the integrated NDT-based LiDAR odometry with GNSS and MGL solutions. The ATE of two trajectories is calculated for comparison.

The comparison of positioning accuracy for the DEC model applied and not applied is shown in Table 1, and the Cumulative Distribution Function (CDF) is shown in Fig. 7. It can be seen from Table 1 that, given the DEC model is applied, all errors, including max, RMSE, Std, and Mean position errors, are reduced significantly. Specifically, when the DEC model is applied, the RMSE error is reduced by 35.5%.

Figure 8 shows how the position error changes over time when the DEC model is applied and not applied, respectively. From the figure, we can see that the positioning error of LiDAR odometry increases over time. The position error of LiDAR odometry decreases significantly once GNSS or MGL solutions are available for fusion, which have typically positioning accuracy of 10-20 cm. However, the error will get accumulated after GNSS and MGL become unavailable, whereas the proposed DEC model can mitigate the accumulated error of the LO.

Table 1 Comparison of positioning accuracy

Solution	RMSE (m)	Max (m)	Std (m)	Mean (m)
DEC applied	1.67	6.28	1.01	1.33
DEC not applied	2.59	9.35	1.73	1.93

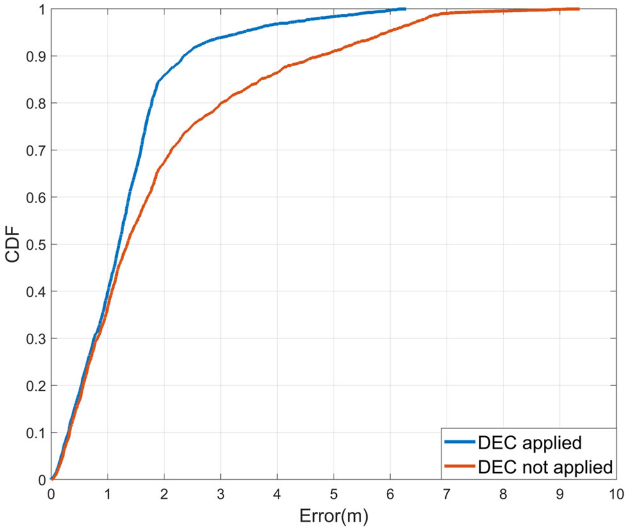


Fig. 7 CDF of positioning error

In this experiment, we prove that the proposed DEC method can effectively reduce the accumulation of position errors of LiDAR odometry when there is no external absolute position for a long time. The performance of the proposed positioning solution will be evaluated in the next experiment.

3.3 Accuracy evaluation for three different scenarios of urban environments

The accuracy evaluation depends on the GNSS-derived ground truth. In order to verify the performance of the proposed solution in urban environments where GNSS may be denied, we simulate three typical scenarios of urban environments as listed in Table 2, and compares the positioning accuracy of different combinations of positioning technologies in these urban scenarios. These typical scenarios are featured with different availability of GNSS and MGL solutions. Scenario A simulates the typical urban environment where GNSS signals are blocked from time to time, and there is no 3D map usable for the MGL method. Thus, GNSS precise positioning can be used at part time, and the LO can be used always. Scenario B simulates the underground space, e.g., tunnel and parking, where GNSS is fully denied, but the 3D map is available for global localization. Thus, the MGL solution is available at a low rate and the LO provides the positions at a high rate. Scenario C simulates the typical urban road environment where GNSS signals are denied partly from time to time, and there are 3D maps intermittently available for the MGL method.

In scenario A, GNSS and LO are fused to provide continuous positioning. We compare the positioning accuracy of four fusion methods: (1) LiDAR odometry only (using the same starting position as the other combinations, provided by GNSS); (2) LiDAR odometry integrated with GNSS directly, which means LiDAR odometry trajectory is aligned with GNSS positions when they are available; (3) LiDAR odometry integrated with GNSS positions using the global graph optimization (GGO), which means the historical trajectory is optimized using GNSS positions as the constraints; (4) LiDAR odometry integrated with GNSS positions and the DEC model is applied to further improve the accuracy. The ATE accuracy comparison of scenario A is shown in Table 3 and Fig. 9.

In scenario B, MGL and LO are fused to provide continuous positioning. We compare the positioning accuracy of four fusion methods: (1) LiDAR odometry only (using the same starting position as the other combinations, provided by

Fig. 8 Position errors between the estimated positions and ground truth

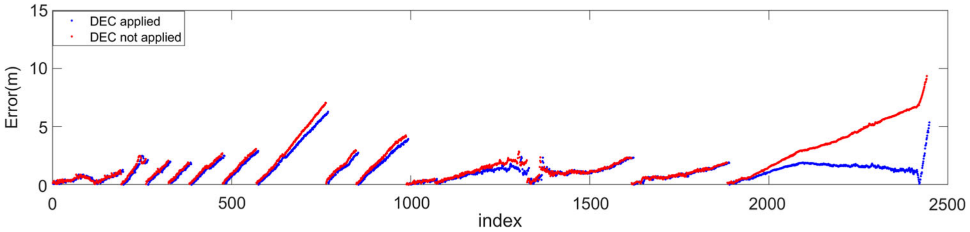


Table 2 Three types of urban scenarios

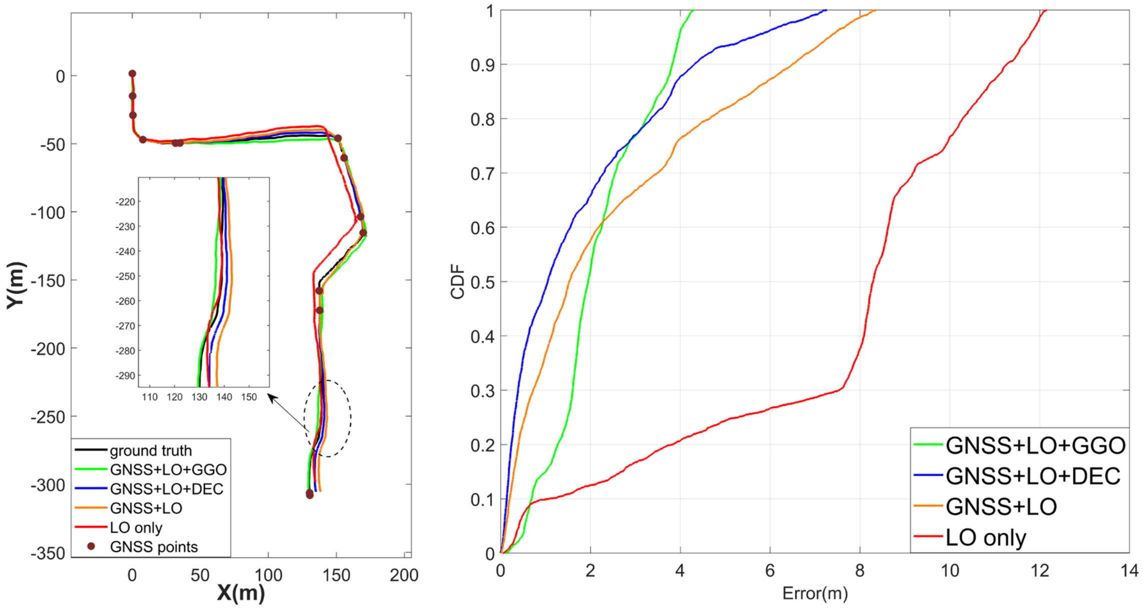
Scenario	Available positioning technologies	Description of simulated scenario types
A	GNSS+LO	GNSS partly denied, without 3D map
B	MGL+LO	GNSS fully denied, partly 3D map available
C	GNSS+MGL+LO	GNSS partly denied, partly 3D map available

Table 3 ATE comparison for scenario A

Solution	Number of fusion solutions	Number of GNSS position	Mean(m)	Std(m)	RMSE(m)	Max(m)
(I)LiDAR odometry only	2445	-	7.42	3.42	8.17	12.16
(II)LO+GNSS	2445	14	2.51	2.32	3.41	8.37
(III)GNSS+LO+GGO	2445	14	2.20	1.01	2.42	4.32
(IV)GNSS+LO+DEC	2445	14	1.78	1.74	2.49	7.26

MGL); (2) LiDAR odometry integrated with MGL directly, which means LiDAR odometry trajectory is aligned with MGL solutions when they are available; (3) LiDAR odometry integrated with MGL solutions using the GGO, which means the historical trajectory is optimized using MGL solutions as the constraints; (4) LiDAR odometry integrated with MGL solutions and the DEC model is applied to further improve the accuracy. The ATE accuracy comparison of scenario B is shown in Table 4 and Fig. 10.

In scenario C, GNSS, MGL and LO are fused to provide continuous positioning. We compare the positioning accuracy of four fusion methods: (1) LiDAR odometry only (using the same starting position as the other combinations, provided by GNSS and MGL); (2) LiDAR odometry inte-

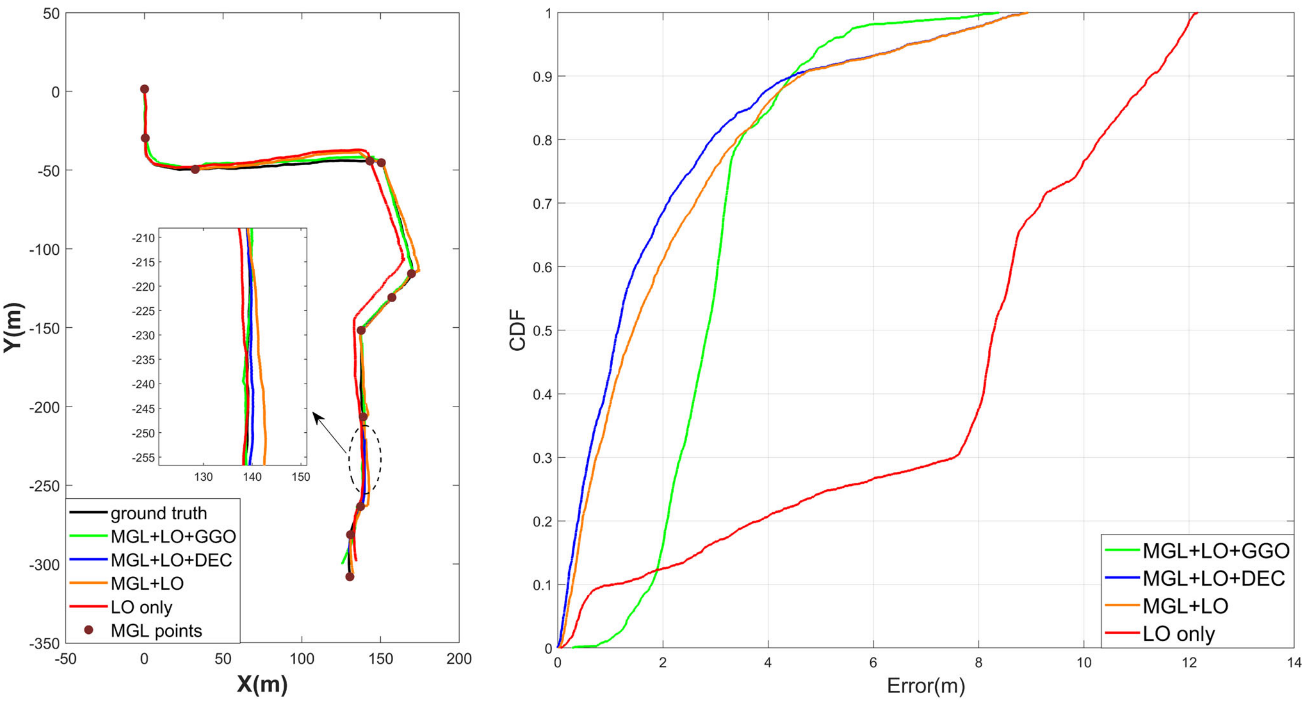


(a) (b)

Fig. 9 Comparison of trajectory **a** and CDF **b** for scenario A

Table 4 ATE comparison for scenario B

Solution	Number of fusion solutions	Number of MGL positions	Mean(m)	Std(m)	RMSE(m)	Max(m)
(I)LiDAR odometry only	2445	-	7.42	3.42	8.17	12.16
(II)LO+MGL	2445	12	2.12	1.96	2.88	8.95
(III)MGL+LO+GGO	2445	12	2.98	1.14	3.19	8.39
(IV)MGL+LO+DEC	2445	12	1.89	1.95	2.71	8.95



(a)

(b)

Fig. 10 Comparison of trajectory **a** CDF and **b** for scenario B

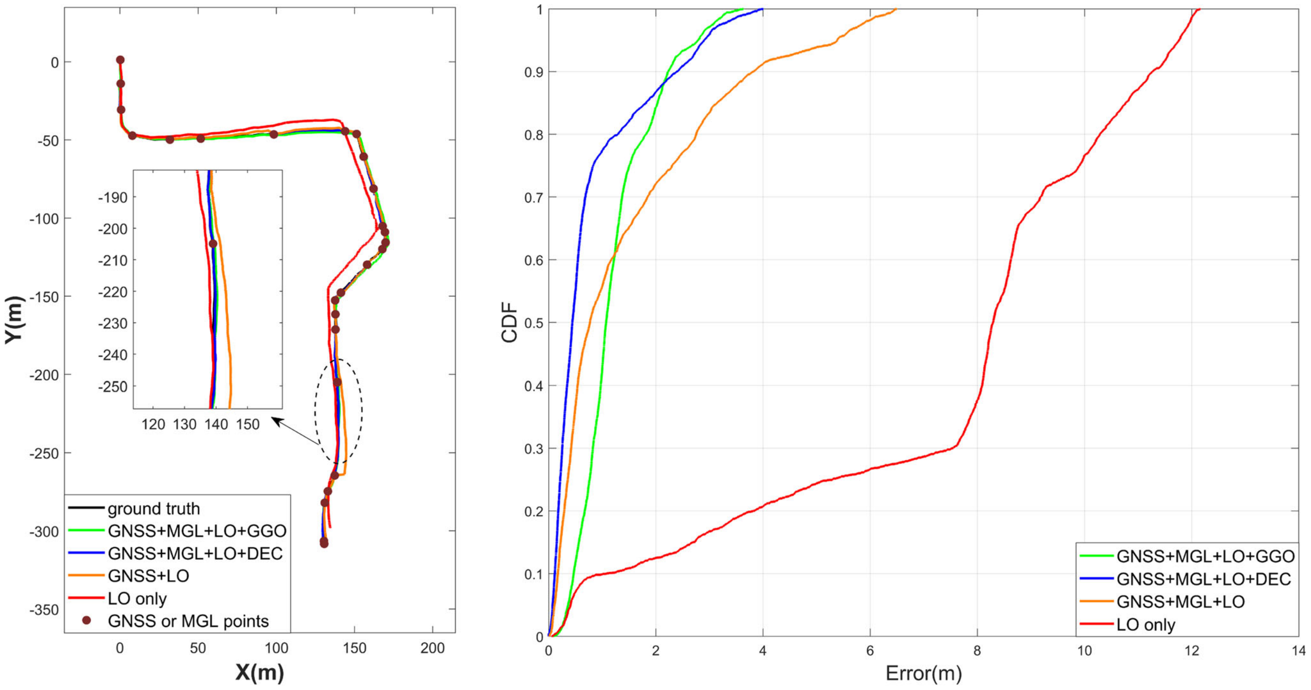
grated with GNSS and MGL directly, which means LiDAR odometry trajectory is aligned with GNSS positions or MGL solutions when they are available; (3) LiDAR odometry integrated with GNSS positions and MGL solutions using the GGO, which means the historical trajectory is optimized using GNSS positions and MGL solutions as the constraints; (4) LiDAR odometry integrated with GNSS positions and MGL solutions and the DEC model is applied to further improve the accuracy. The ATE accuracy comparison of scenario C is shown in Table 5 and Fig. 11.

By comparing these results of three scenarios, although the availability of GNSS and MGL solutions is limited in urban environments, the integrated solution has the availability of 100%, thanks to the continuity of the LO. Here the dataset contains 2445 LiDAR scans, and it has the same number of positioning solutions. The limited ($<1\%$) number of GNSS and MGL solutions are advantageous for improving the accuracy of the proposed fusion positioning solution. Compared to the LO, the graph optimization-based fusion solution improves the accuracy of historical trajectory by 60.9%−82.6% for different scenarios, and the proposed DEC model improves the accuracy of real-time trajectory by 66.8-−85.2%, in terms of the mean error. The comparison of Tables 3, 4 and 5 shows that more absolute positioning solutions (e.g., either GNSS or MGL) can improve further the positioning accuracy as there are more opportunities for global optimization. The experimental results show that the positioning accuracy of scenario C is improved by 51.3% and 55.3% (RMSE), compared to scenarios A and B, respectively.

From the CDF comparison of four trajectories in each scenario, the positioning accuracy of LiDAR odometry integrated with GNSS by the proposed DEC (Solution IV) is higher than other positioning methods. According to Tables 3, 4 and 5, we can find that the max positioning error of the fusion method with the GGO (Solution III) is smaller than the fusion method with the proposed DEC (Solution IV). It should be noted that the GGO optimized all historical measurements with an objective of the minimum overall error,

Table 5 ATE comparison for scenario C

Solution	Number of fusion solutions	Number of MGL position	Mean(m)	Std(m)	RMSE(m)	Max(m)
(I)LiDAR odometry only	2445	-	7.42	3.42	8.17	12.16
(II)LO+GNSS +MGL	2445	26	1.54	1.55	2.19	6.50
(III)LO+GNSS +MGL+GGO	2445	26	1.24	0.70	1.42	3.64
(IV)LO+GNSS +MGL+DEC	2445	26	0.86	0.86	1.21	4.01



(a)

(b)

Fig. 11 Comparison of trajectory **a** CDF **b** for scenario C

and the derived trajectory is used as the reference to calculate the DEC model, while Solution IV provides the real-time trajectory that corrects the LO using the previously derived DEC model. In other words, Solution III provides the globally optimized reference trajectory when absolute positions are available, and hence it has latency. Solution IV provides real-time positioning solutions, and it is more suitable for applications that require high real-time performance, such as autonomous driving. The result obtained by fusing GNSS and MGL through graph optimization is not better than direct fusion in scenario B, because there are fewer available absolute positions. The graph optimization may result in a larger error locally, in order to obtain the global optimal. However, the max error of Solution III is still smaller than that of Solution II.

4 Discussion and Conclusion

In this work, we proposed a GNSS/LiDAR/Map integrated positioning solution for ubiquitous positioning in urban environments. The proposed solution integrates multiple positioning technologies, including GNSS precise positioning, 3D map matching-based global localization, and LiDAR odometry, which are applicable in the autonomous driving context. Our proposed fusion method combines the advantages of absolute positioning and relative positioning, and resultingly the fusion solution has 100% availability and the positioning accuracy is improved significantly. Furthermore, considering the error characteristics of the LO technology, we propose a drift error correction model. The proposed drift error model and integrated positioning solution are validated using the NCLT open dataset and the corresponding ground truth.

In our first experiments, we tested the effectiveness of the DEC model for reducing the error of the LiDAR odometry. The experiment results show that the DEC model improves the accuracy of the LO by 35.5%. The proposed DEC can also be used to improve the positioning accuracy of other LiDAR odometry methods because that DEC only corrects the results of LiDAR odometry and does not involve LiDAR scan matching. In order to evaluate the performance of the proposed solution, we simulate three types of urban scenarios, where these positioning technologies (LO, MGL, and GNSS) have different availability. Thanks to the continuity of the LO, all fusion methods provide 100% availability of integrated solutions. Precise absolute positioning solutions can be used to improve the accuracy of historical trajectory using the graph optimization method, and the historical trajectory is then used for updating online the DEC model. The DEC model is used for improving the accuracy of the integrated solution, especially when precise absolute positioning solutions are not available for a long time. Compared to the LO, the fusion solution with the proposed DEC model improves the accuracy of the real-time trajectory by 66.8%−85.2%, depending on the availability of absolute positioning solutions. To verify the performance of the proposed DEC model and the fusion method, we use only very limited (roughly 1%) absolute positioning solutions. In scenario C, both GNSS and MGL can be used to calculate the error model of LiDAR odometry and decrease LiDAR odometry drift error. So, this scenario achieves the best localization results compared to scenarios A and B, due to the increase in available absolute positions. Our experiment results show that the proposed positioning solution can provide precise and continuous positioning for autonomous vehicles in real-time in urban.

The applicability of the linear DEC model proposed in this study is limited, especially when the scanning environment has significant change. For example, the LO error may follow a different pattern when the heading has large turning, or the surrounding geometric structure changes significantly. More sophisticated models, e.g., deep neural networks-based methods, will be studied in the future to improve the accuracy and applicability of the DEC model. In addition, the NDT-based LiDAR odometry gets relative position only by scan-to-scan matching, an advanced LiDAR odometry with better accuracy will used in our positioning solution.

Acknowledgements This study was supported in part by National Key Research Development Program of China with project No. 2021YFB2501102, the Natural Science Fund of China with Project No. 41874031 and 42111530064, Shenzhen Science and Technology Program with Project No. JCYJ20210324123611032, Wuhan Knowledge Innovation Research Program with Project No. 2022010801010109.

Author Contributions Jingbin Liu designed this research; Jingbin Liu and Yifan Liang designed the experiments; Yifan Liang, Dong Xu and Xiaodong Gong wrote the program code and performed the experiments; Jingbin Liu and Yifan Liang wrote the paper; Juha Hyyppä edited the manuscript.

Data Availability The NCLT datasets used in the current study are available from http://robots.engin.umich.edu/nclt/.

References

Bhamidipati S, Gao GX (2020) Integrity monitoring of graph-slam using gps and fish-eye camera. Navigation 67(3):583–600. https://doi.org/10.1002/navi.381

Bosse M, Zlot R (2013) Place recognition using keypoint voting in large 3d lidar datasets. In: IEEE International Conference on Robotics & Automation

Carlevaris-Bianco N, Ushani AK, Eustice RM (2016) University of michigan north campus long-term vision and lidar dataset. Int J Robotics Res 35(9):1023–1035. https://doi.org/10.1177/0278364915614638

Chang L, Niu X, Liu T et al (2019) Gnss/ins/lidar-slam integrated navigation system based on graph optimization. Remote Sensing 11(9):1009. https://doi.org/10.3390/rs11091009

Chen S (2020) Research on slam based on lidar/visual fusion (lv-slam). doctor, https://doi.org/10.27379/d.cnki.gwhdu.2020.000697

Elbaz G, Avraham T, Fischer A (2017) 3d point cloud registration for localization using a deep neural network auto-encoder. 30th Ieee Conference on Computer Vision and Pattern Recognition (Cvpr 2017) pp 2472–2481. https://doi.org/10.1109/Cvpr.2017.265, <GotoISI>://WOS:000418371402057 https://ieeexplore.ieee.org/stampPDF/getPDF.jsp?tp=&arnumber=8099748&ref=

Finman R, Paull L, Leonard JJ (2015) Toward object-based place recognition in dense rgb-d maps. In: ICRA Workshop Visual Place Recognition in Changing Environments, Seattle, WA

Gao Y, Liu S, Atia MM, et al (2015) Ins/gps/lidar integrated navigation system for urban and indoor environments using hybrid scan matching algorithm. Sensors (Basel) 15(9):23,286–302. https://doi.org/10.3390/s150923286, https://www.ncbi.nlm.nih.gov/pubmed/26389906

Grisetti G, Stachniss C, Burgard W (2007) Improved techniques for grid mapping with rao-blackwellized particle filters. Ieee Transactions on Robotics 23(1):34–46. https://doi.org/10.1109/Tro.2006.889486 <GotoISI>://WOS:000244311500004

Grisetti G, Kummerle R, Stachniss C et al (2010) A tutorial on graph-based slam. IEEE Intell Transp Syst Magazine 2(4):31–43. https://doi.org/10.1109/mits.2010.939925

He G, Yuan X, Zhuang Y et al (2021) An integrated gnss/lidar-slam pose estimation framework for large-scale map building in partially gnss-denied environments. IEEE Trans Instrum Measure 70:1–9. https://doi.org/10.1109/tim.2020.3024405

Hess W, Kohler D, Rapp H, et al (2016) Real-time loop closure in 2d lidar slam. In: 2016 IEEE International Conference on Robotics and Automation (ICRA). IEEE, pp 1271–1278

Holz D, Ichim A, Tombari F et al (2015) Registration with the point cloud library - a modular framework for aligning in 3-d. IEEE Robotics Autom Mag 22(4):110–124

Ji Z, Singh S (2014) Loam: Lidar odometry and mapping in real-time. In: Robotics: Science and Systems Conference

Joerger M, Spenko M (2017) Towards navigation safety for autonomous cars. Inside GNSS

Julier SJ, Durrant-Whyte HF (2003) On the role of process models in autonomous land vehicle navigation systems. IEEE Trans Robot Autom 19:1–14. https://doi.org/10.1109/TRA.2002.805661

Kim J, Sukkarieh S (2005) 6dof slam aided gnss/ins navigation in gnss denied and unknown environments. J Global Position Syst 4(1&2):120–128. https://doi.org/10.5081/jgps.4.1.120

Kohlbrecher S, Stryk OV, Meyer J, et al (2011) A flexible and scalable slam system with full 3d motion estimation. In: IEEE International Symposium on Safety

Kukko A, Kaijaluoto R, Kaartinen H et al (2017) Graph slam correction for single scanner mls forest data under boreal forest canopy. ISPRS J Photogramm Remote Sensing 132:199–209. https://doi.org/10.1016/j.isprsjprs.2017.09.006

Li P, Wang R, Wang Y et al (2020) Evaluation of the icp algorithm in 3d point cloud registration. IEEE Access 8:68. https://doi.org/10.1109/access.2020.2986470

Li W, Cui X, Lu M (2020b) Feature-based tightly-integrated rtk/ins/lidar fusion positioning algorithm in ambiguity domain. In: China Satellite Navigation Conference, Springer, pp 510–526

Li Y, Zahran S, Zhuang Y et al (2019) Imu/magnetometer/barometer/mass-flow sensor integrated indoor quadrotor uav localization with robust velocity updates. Remote Sensing 11(7):838

Lu W, Wan G, Zhou Y, et al (2019) Deepicp: An end-to-end deep neural network for 3d point cloud registration. arxiv 2019. http://arxiv.org/abs/1905.04153

Lu W, Zhou Y, Wan G, et al (2020) L3-net: Towards learning based lidar localization for autonomous driving. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)

Ma H, Yin DY, Liu JB et al (2022) 3d convolutional auto-encoder based multi-scale feature extraction for point cloud registration. Optics & Laser Technol 149:78. https://doi.org/10.1016/j.optlastec.2022.107860

Magnusson M (2009) The three-dimensional normal-distributions transform:an efficient representation for registration, surface analysis, and loop detection. PhD thesis, orebro university

Novatel (2022) https://novatel.com/

Pierzchała M, Giguère P, Astrup R (2018) Mapping forests using an unmanned ground vehicle with 3d lidar and graph-slam. Computers Electron Agric 145:217–225. https://doi.org/10.1016/j.compag.2017.12.034

Qian C, Zhang H, Li W et al (2020) A lidar aiding ambiguity resolution method using fuzzy one-to-many feature matching. J Geodesy 94:10. https://doi.org/10.1007/s00190-020-01426-z

Rusu RB, Blodow N, Beetz M (2009) Fast point feature histograms (fpfh) for 3d registration. Icra: 2009 Ieee International Conference on Robotics and Automation, Vols 1-7 pp 1848–1853. <GotoISI>://WOS:000276080400298

Schultz A, Gilabert R, Bharadwaj A, et al (2016) A navigation and mapping method for uas during under-the-canopy forest operations. In: 2016 IEEE/ION Position, Location and Navigation Symposium - PLANS 2016

Shamsudin AU, Ohno K, Hamada R et al (2018) Consistent map building in petrochemical complexes for firefighter robots using slam based on gps and lidar. ROBOMECH J 5:1. https://doi.org/10.1186/s40648-018-0104-z

Shan T, Englot B (2019) Lego-loam: Lightweight and ground-optimized lidar odometry and mapping on variable terrain. In: 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)

Shan T, Englot B, Meyers D, et al (2020) Lio-sam: Tightly-coupled lidar inertial odometry via smoothing and mapping. In: 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, pp 5135–5142

Steder B, Grisetti G, Burgard W (2010) Robust place recognition for 3d range data based on point features. 2010 Ieee International Conference on Robotics and Automation (Icra) pp 1400–1405. https://doi.org/10.1109/Robot.2010.5509401, <GotoISI>://WOS:000284150001132 https://ieeexplore.ieee.org/stampPDF/getPDF.jsp?tp=&arnumber=5509401&ref=

Stephenson S, Meng X, Moore T, et al (2011) Accuracy requirements and benchmarking position solutions for intelligent transportation location based services. In: Proceedings of the 8th international symposium on location-based services

Wang H, Wang C, Chen CL, et al (2021) F-loam : Fast lidar odometry and mapping. 2021 Ieee/Rsj International Conference on Intelligent Robots and Systems (Iros) pp 4390–4396. https://doi.org/10.1109/Iros51168.2021.9636655, <GotoISI>://WOS:000755125503067 https://ieeexplore.ieee.org/stampPDF/getPDF.jsp?tp=&arnumber=9636655&ref=

Wang ZM, Zhang Q, Li JS, et al (2019) A computationally efficient semantic slam solution for dynamic scenes. Remote Sensing 11(11). https://doi.org/10.3390/rs11111363, https://mdpi-res.com/d_attachment/remotesensing/remotesensing-11-01363/article_deploy/remotesensing-11-01363.pdf?version=1559823029

West KF, Webb BN, Lersch JR, et al (2004) Context-driven automated target detection in 3D data. In: Sadjadi FA (ed) Automatic Target Recognition XIV, International Society for Optics and Photonics, vol 5426. SPIE, pp 133 – 143, https://doi.org/10.1117/12.542536

Xu D, Liu J, Hyyppä J et al (2022) A heterogeneous 3d map-based place recognition solution using virtual lidar and a polar grid height coding image descriptor. ISPRS J Photogram Remote Sensing 183:1–18. https://doi.org/10.1016/j.isprsjprs.2021.10.020

Xu D, Liu J, Liang Y et al (2022) A lidar-based single-shot global localization solution using a cross-section shape context descriptor. ISPRS J Photogramm Remote Sensing 189:272–288. https://doi.org/10.1016/j.isprsjprs.2022.05.005

Yurtsever E, Lambert J, Carballo A et al (2020) A survey of autonomous driving: Common practices and emerging technologies. IEEE Access 8:58443–58469

Zeng A, Song SR, Niessner M, et al (2017) 3dmatch: Learning local geometric descriptors from rgb-d reconstructions. 30th Ieee Conference on Computer Vision and Pattern Recognition (Cvpr 2017) pp 199–208. https://doi.org/10.1109/Cvpr.2017.29, <GotoISI>://WOS:000418371400022 https://ieeexplore.ieee.org/stampPDF/getPDF.jsp?tp=&arnumber=8099512&ref=

Zhang J, Singh S (2017) Low-drift and real-time lidar odometry and mapping. Autonomous Robots 41(2):401–416