IEEE Open Journal of
Intelligent Transportation Systems

Received 29 March 2024; revised 23 May 2024; accepted 23 May 2024. Date of publication 28 May 2024; date of current version 18 June 2024.

Digital Object Identifier 10.1109/OJITS.2024.3406390

LiDAR and IMU Tightly Coupled Localization System Based on Ground Constraint in Flat Scenario

MAN YU[1], KEYANG GONG[2], WEIHUA ZHAO[3,4], AND RUI LIU[5]

[1]School of Construction Machinery, Chang'an University, Xi'an 710064, China
[2]Intelligent Driving Center, Geely Automotive Research Institute (Ningbo) Company, Ningbo 315336, China
[3]School of Vehicle Engineering, Xi'an Aeronautical Institute, Xi'an 710061, China
[4]Key Laboratory of Operation Safety Technology on Transport Vehicles, Ministry of Transport, Beijing 100088, China
[5]School of Automobile, Chang'an University, Xi'an 710018, China

CORRESPONDING AUTHOR: R. LIU (e-mail: liuruiaza@163.com)

This work was supported in part by the Key Research and Development Program of Shaanxi Province under Grant 2023-YBGY-122; in part by the Opening Project of the Key Laboratory of Operation Safety Technology on Transport Vehicles under Grant KFKT2018-05; and in part by the Natural Science Basic Research Program of Shaanxi under Grant 2022JQ-599 and Grant 2022JQ-543.

ABSTRACT Accurate estimation of current position and attitude of a vehicle is one of the key technologies for autonomous driving. Due to the defect of LiDAR intrinsic parameter and the sparsity of LiDAR beam in the vertical direction, current LiDAR-based simultaneous localization and mapping (SLAM) system generally suffers from the problem of inaccurate height positioning. In this study, a LiDAR and inertial measurement unit (IMU) tightly coupled localization algorithm considering ground constraint is proposed, which is developed based on a pose graph optimization framework. At the front end, the ground segmentation algorithm Patchwork is improved to obtain a point cloud with higher verticality, which is added to the LiDAR inertial odometry. Moreover, constraints are constructed by using current frame ground points and world map ground points, which are added to factor map optimization to limit elevation errors. At the back end, SC++ descriptors are used to construct loop constraints to eliminate accumulated errors. Verifications based on KITTI dataset show that the height positioning accuracy will be improved through introducing ground constraint factor and loop detection factor. Real vehicle tests indicate that the proposed algorithm has better height positioning accuracy and better robustness compared with the LeGO-LOAM algorithm.

INDEX TERMS LiDAR inertial system, height positioning, pose graph optimization, ground constraint.

I. INTRODUCTION

SIMULTANEOUS localization and mapping (SLAM) plays a crucial role in autonomous driving. Results of SLAM algorithms can be considered as perception of intelligent vehicle, which is the basis for decision, planning, and control of autonomous vehicle [1], [2], [3]. SLAM can be realized with a variety of sensors, including cameras [4], radar [5], [6] and LiDAR [7], [8]. Camera-based visual SLAM algorithms, including ORB-SLAM [9], [10], VINS-Mono [11], are sensitive to the illumination and is affected by motion blur when driving speed is high. LiDAR can provide precise information about the surrounding environment [12]. Therefore, LiDAR and IMU tightly coupled SLAM becomes the main localization method in autonomous driving [13], [14], [15], [16].

Nevertheless, influences of measurement noises on LiDAR-based SLAM have not received enough attention. It was found that LiDAR had a high measurement bias when incidence angles were high, which would result in a slight curvature of observed points on the road surface [17]. Moreover, the sparsity of LiDAR beam in the vertical direction made LiDAR had insufficient vertical resolution. What's more, dew, artificial dirt, and foam would affect output data of LiDAR beam [18], and information loss would take place during vertical vibration [19]. Therefore, results of SLAM are easily to drift along the vertical direction. In

The review of this article was arranged by Associate Editor Xin Xia.

this paper, a LiDAR-inertial localization system considering ground constraint is proposed to increase accuracy of height positioning. The ground constraint is applied to reduce the error caused by LiDAR measurement bias in a flat scenario. Main contributions of this paper are as follows:

i) A SLAM framework based on pose graph optimization integrates the ground segmentation algorithm at the front end to effectively separate the ground points with higher verticality.

ii) To convert the plane expressed in Hesse Form (HF) form into a three-dimensional vector through a unique projection method to avoid over-parameter phenomenon in the optimization process.

iii) Construct the ground constraint factor through the current frame ground points and the world map ground points, and calculate the residual and Jacobian matrix.

II. RELATED WORK

The ground segmentation methods are studied by many researchers [20], [21], which is commonly used for two purposes, one is for drivable areas extraction in navigation [22], and the other is for object tracking or target recognition [23], [24]. Because most of targets that have interaction with the host vehicle are road users in contact with the ground, these road users can be easily identified with a low computational cost when ground points are eliminated [25]. Patchwork is one of the most important ground segmentation algorithms for the purpose of traffic target identification [26]. It is validated in KITTI database and results show that it has little performance disturbance and strong robustness in complex and uneven urban roads. However, Patchwork aims to eliminate dynamic objects on the ground, which also regards bulges and potholes on the road as ground points. Hence, it is not suitable for adding ground constraint in SLAM when higher planeness of ground points is required. Consequently, current ground segmentation algorithms are mainly aimed at drivable area extraction and target tracking, so they do not consider the requirement of vertical constraints. Therefore, Patchwork is improved in this paper to acquire ground with higher planeness, which can provide vertical constraints to reduce vertical drift of the localization algorithm.

For the representation of a plane, three methods are mainly employed. The first one is Hesse form (HF), which is represented by the normal vector of the plane and the distance from the origin of the coordinate system to the plane. Because a three-dimensional vector is utilized to represent two degrees of freedom, singular matrixes may appear when performing least squares optimization. The second is to use the azimuth and elevation angles to represent the normal vector in spherical coordinates, but ambiguity will emerge when the elevation angle is $\pm\pi/2$. The third is to use the unit quaternion to represent the plane [27], but the physical connection between the quaternion and the plane is unknown.

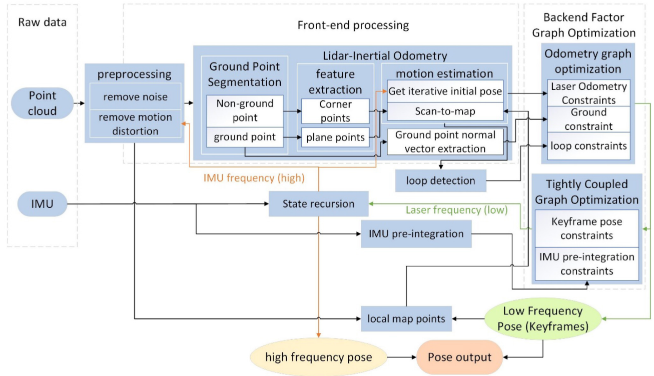


FIGURE 1. Block diagram of laser inertial SLAM system based on ground constraint.

The idea of using ground points is reflected in some typical SLAM frameworks. For example, ground points in LeGO-LOAM [28] and LOAM [29] are extracted to estimate 3 of the 6 degrees of freedom, but ground information is not integrated into the pose graph optimization framework in these works. Ground-SLAM introduces ground constraint in pose graph optimization to reduce errors in roll angle, pitch angle, and vertical displacement [30]. However, RANSAC algorithm is applied in the ground segmentation in Ground-SLAM, which has a large amount of calculation. Moreover, it constructs constraints through ground points extracted between adjacent frames, which has poor stability.

III. LIDAR-INERTIAL LOCALIZATION SYSTEM BASED ON GROUND CONSTRAINT

First, the proposed localization system overview is introduced. Next, the front end processing is indicated. In the end, the back end optimization is demonstrated.

A. SYSTEM OVERVIEW

The system block diagram of the localization and mapping algorithm based on the tight coupling of LiDAR and IMU based on ground constraint is shown in Fig. 1, which mainly includes the following parts: one is front-end processing, including ground point segmentation, feature extraction, LiDAR inertial odometry, etc. The second is the optimization of the back end factor graph, which mainly includes the construction of the odometry factor, the ground point constraint factor and the loop constraint factor. The main algorithm flow of the system is:

i) For the point cloud data of the current frame, motion distortion is recursively removed through the optimized pose and IMU state of the previous frame, and then passed to the front-end for processing.

ii) A ground point segmentation algorithm is used to divide the point cloud into ground points and non-ground points.

iii) Use the curvature attribute near the point to uniformly extract corner points and plane points, and calculate the normal vector of the ground point fitting plane.

iv) For the historical frame, the point cloud in the LiDAR coordinate system is transferred to the world coordinate system to obtain world map points through the optimized pose, and the current frame is used to match the world map, then the L-M algorithm is used to solve the odometry pose of the current frame whose iteration initial value is obtained by IMU integration.

v) The normal vector of the current frame ground points and the normal vector of the world map ground points constitute a plane constraint. The loop closure detection is run through a separate thread, and the loop closure constraint is obtained when a loop closure is detected. The odometry constraint, ground constraint, and loop closure constraint are added to the factor graph optimization to obtain a more accurate global pose.

vi) The low-frequency pose output by the front-end and the IMU pre-integration constraints are added to the optimization of the tightly coupled factor graph, and the iterative initial value for the world map matching of the next LiDAR frame is obtained through optimization.

vii) In the end, the SLAM system can output high-frequency and low-frequency poses. Through the poses, the point cloud collected by each frame of radar can be transformed into a global map in the world coordinate system, thereby completing the localization and mapping process.

B. FRONT END PROCESSING

The front end processing includes two parts, i.e., ground point segmentation and motion estimation.

1) GROUND POINT SEGMENTATION

The proposed ground segmentation algorithm, which is named AdationBin, is mainly composed of ground point extraction based on bird-eye view and outlier point elimination based on likelihood function.

1) Mesh region division:The bird-eye view [31] is applied to divide a point cloud frame into mesh grids. Adaptive mesh division method is adopted to avoid the lack of representation caused by small mesh area the distance is too close or the sparse phenomenon caused by too few points in a grid when the distance is far. The bird eye view is divided into four areas according to the distance, which is denoted as

$$Q_n = \left\{\boldsymbol{p}_h \in P | L_{\min,n} \le \rho_h \le L_{\max,n}\right\}, (n = 1, 2, 3, 4)$$

where P is a frame of point cloud. $\boldsymbol{p}_h$ is the h-th point, whose coordinate is $[x_h, y_h, z_h]^{\mathrm{T}}$. $\rho_h = \sqrt{{x_h}^2 + {y_h}^2}$. $L_{\min,n}$ is the inner boundary of the n-th area, and $L_{\max,n}$ is the outer boundary of the n-th area. Each frame of point cloud is divided into $N_{\rho,n} \times N_{\theta,n}$ grids according to distance and azimuth, and point set in each grid $S_{p,q,n}$ is defined as follows

$$S_{\boldsymbol{p},\boldsymbol{q},\boldsymbol{n}} = \left\{ \begin{array}{l} \boldsymbol{p}_h \in Q_n | \frac{(p-1)\cdot L_n}{N_{\boldsymbol{\rho},\boldsymbol{n}}} \le \rho_h - L_{\min,n} < \frac{p\cdot L_n}{N_{\boldsymbol{\rho},\boldsymbol{n}}}, \\ \frac{(q-1)\cdot 2\pi}{N_{\boldsymbol{\theta},\boldsymbol{m}}} \le \theta_h + \pi < \frac{q\cdot 2\pi}{N_{\boldsymbol{\theta},\boldsymbol{m}}} \end{array} \right\} \quad (1)$$

where $\theta_h = \arctan(y_h/x_h)$, $L_n = L_{\max,\boldsymbol{n}} - L_{\min,\boldsymbol{n}}$, $L_{\max,\boldsymbol{n}} = L_{\min,\boldsymbol{n}+1}$, $L_{\max} = L_{\max,4}$, $L_{\min} = L_{\min,1}$, $L_{\min,2} = \frac{6L_{\min}+L_{\max}}{7}$, $L_{\min,3} = \frac{(4L_{\min}+L_{\max})}{5}$, and $L_{\min,4} = \frac{2L_{\min}+L_{\max}}{3}$.

Larger grids are utilized in areas Q_1 and Q_4 to mitigate the lack of representation and the sparsity deficiencies mentioned above.

2) ground-point-extraction:

Ground points are extracted after grid division. First, initial ground points are obtained based on z-coordinates of points

$$\widehat{G}_m^{(0)} = \left\{\boldsymbol{p}_h \in S_m | z\left(\boldsymbol{p}_h\right) < z_{\text{mean}} + z_{\text{thr}}\right\} \quad (2)$$

where $\widehat{G}_m^{(0)}$ indicates initial ground points contained in the m-th grid. $z(\cdot)$ indicates the z-coordinate of the point, z_{mean} is the average value of z-coordinate in one grid, and z_{thr} is the ground point threshold in the vertical direction.

Second, the ground point plane is iteratively optimized. Principal components analysis (PCA) algorithm is adopted [32] because the commonly used RANSAC algorithm is time-consuming [33]. In the PCA algorithm, a covariance matrix $C \in \mathbb{R}\sim^{3\times 3}$ for points in a grid is calculated. The eigenvalues of the matrix from large to small are denoted as λ_1, λ_2, λ_3, and the corresponding eigenvectors are denoted as $\boldsymbol{v}_1$, $\boldsymbol{v}_2$, $\boldsymbol{v}_3$, respectively. The normal vector of the ground point plane can be denoted as $\boldsymbol{n}=\boldsymbol{v}_3$, and therefore the plane coefficient can be expressed as $d=-\boldsymbol{n}\cdot\boldsymbol{p}_{\text{med}}$, where $\boldsymbol{p}_{\text{med}}$ is the median point in the grid. The difference of this plane coefficient calculated by two adjacent iterations less than a certain threshold is employed as a termination criterion of the iteration. Hence, the iterative can be denoted as

$$\widehat{G}_m^{(l+1)} = \left\{\boldsymbol{p}_h \in S_m | {d_m}^{(l)} - \widehat{d}_h < d_{\text{thr}}\right\} \quad (3)$$

where $\widehat{G}_m^{(l)}$ indicates ground points contained in the m-th grid after the l-th iteration. $\widehat{d}_h = -{\boldsymbol{n}_m}^{(l)} \cdot \boldsymbol{p}_h$. d_{thr} is the iterative threshold, which is set to be a constant.

3) outliers-elimination:

Bulges and holes on the ground are eliminated as outliers by using a binary classification method based on the area probability detection. Let $L(\theta|\boldsymbol{X})$ be the likelihood function. θ represents all parameters in a probability distribution, which can be understood as characteristics of ground points. $\boldsymbol{X}$ represents the random variable, which is expressed as a ground point or a non-ground point. Assuming the grids are independent of each other, the likelihood function is expressed as

$$L(\theta|\boldsymbol{X}) = f(\boldsymbol{X}|\theta) = \prod_m f(\boldsymbol{X}_m|\theta_m) \quad (4)$$

where $\boldsymbol{X}_m$ and θ_m represent variables and grid parameters of the m-th grid.

Vertical drift can be suppressed by using ground point segmentation with higher planeness. In order to increase the planeness of the ground constraints, the uprightness and the elevation in Patchwork are improved to obtain better horizontal ground constraints. Characteristics of ground points can be described by verticality, height and smoothness [26]. The smoothness is not considered here because it is introduced to restore the wrongly excluded slope point. Therefore, the probability density function of the grid is expressed as

$$f(\boldsymbol{X}_m|\theta_m) \triangleq \varphi\left(\boldsymbol{v}_{3,m}\right)\psi(z_{\text{mean}}, s_m) \tag{5}$$

where $\varphi(\boldsymbol{v}_{3,m})$ is the verticality probability density function, and $\psi(z_{\text{mean}}, s_m)$ is the height probability density function.

Bulges and holes on the ground will be regarded as outliers because the proposed ground extraction method has high requirements for the verticality and height of ground points, so probability density functions are expressed as

$$\varphi\left(\boldsymbol{v}_{3,m}\right) = \begin{cases} \frac{\boldsymbol{v}_{3,m}\cdot \boldsymbol{z}}{\|\boldsymbol{v}_{3,m}\|\|\boldsymbol{z}\|} > p_{\text{thr}} \\ 0, \quad \text{else} \end{cases} \tag{6}$$

$$\psi(z_{\text{mean}}, s_m) = \begin{cases} \left(1 + e^{-z_{\text{mean}}+k(s_m)}\right)^{-1}, s_m < L_{\text{thr}} \text{ and } z_{\text{mean}} < h_{\text{thr}} \\ \left(1 + e^{z_{\text{mean}}-k(s_m)}\right)^{-1} s_m < L_{\text{thr}} \text{ and } z_{\text{mean}} > h_{\text{thr}} \\ 1, \quad \text{else} \end{cases} \tag{7}$$

where $\boldsymbol{z}$=[0, 0, 1]$^{\text{T}}$. p_{thr} is the perpendicularity threshold, generally set to 0.8∼0.98. $k(s_m)$ is the height threshold that changes with the area. L_{thr} is the range threshold, which means that filtering according to height threshold will not be performed on points outside a certain radius. h_{thr} means the height threshold for pothole elimination, which is related to the installation height of the LiDAR.

The final estimated ground points are represented as follows

$$\widehat{G} = \bigcup_{m\in M} \left[f(X_m|\theta_m) > 0.5\right] \widehat{G}_m \tag{8}$$

where M is the number of grids in a frame of point cloud, and $[\cdot]$ is Iverson bracket.

2) MOTION ESTIMATION

A matching algorithm based on the feature points to the world map (scan-to-map) is used for motion estimation. Feature points extraction method is inherited from LOAM, i.e., plane points S_p and the corner points S_e are distinguished according to the curvature. Plane points are those with curvature less than a plane threshold, whereas corner points are those with curvature greater than a corner threshold.

In the current point cloud frame, the pose of LiDAR is represented by a transformation matrix ($\boldsymbol{T}_{wLc}$), and coordinates of the i-th point in plane points and the j-th point in corner points are denoted as $\boldsymbol{X}_{c,e,i}$ and $\boldsymbol{X}_{c,p,j}$. Therefore, points in the world coordinate system can be obtained through $\boldsymbol{T}_{wLc}$. The distance residuals are calculated by using point-line iterative closest point (ICP) algorithm and point-plane ICP algorithm, which can be expressed as

$$\begin{cases} d_e = f_e\left(\boldsymbol{X}_{c.e,i}, \boldsymbol{T}_{wLk}\right), i \in S_e \\ d_p = f_p\left(\boldsymbol{X}_{c,p,j}, \boldsymbol{T}_{wLk}\right), \boldsymbol{j} \in S_p \end{cases} \tag{9}$$

Thus, an optimization as equation (10) can be achieved and can be solved iteratively by using Levenberg-Marquardt algorithm.

$$(\boldsymbol{T}_{wLk})^* = \arg\min_{\boldsymbol{T}_{wLk}} \left(\sum_i \left\|d_{e,i}\right\|^2 + \sum_j \left\|d_{p,\boldsymbol{j}}\right\|^2\right) \tag{10}$$

C. BACK END FACTOR GRAPH OPTIMIZATION

The back end optimization includes three parts, i.e., ground constraint factor, loop constraint factor and factor graph model.

1) GROUND CONSTRAINT FACTOR

Extracted ground points provide position constraints in z-axis as well as rotational constraints in roll and pitch for back end optimization, i.e., ground constraint.

1) residual-calculation:

The ground constraint is constructed by matching ground points in the current frame with those in the world map. This ground constraint is a binary constraint, that is, poses and ground points in world map can be optimized at the same time. Ground points are expressed in the form of Hesse Form (HF). Therefore, ground points in the current frame and in the world map are expressed as $\boldsymbol{\Omega}_c$=[$\boldsymbol{n}_c$,d_c]$^{\text{T}}$ and $\boldsymbol{\Omega}_w$=[$\boldsymbol{n}_w$,d_w]$^{\text{T}}$, respectively. The normal vector in the current frame $\boldsymbol{n}_c$=[n_{cx},n_{cy},n_{cz}]$^{\text{T}}$ and the normal vector in the world map $\boldsymbol{n}_w$=[n_{wx},n_{wy},n_{wz}]$^{\text{T}}$ are calculated using the PCA algorithm. The attitude of LiDAR ($\boldsymbol{R}_{wLc}$) is utilized to transfer the normal vector in the world map to a transformed normal vector in the current frame

$$\boldsymbol{n}_t = \boldsymbol{R}_{wLc}{}^T\boldsymbol{n}_w \tag{11}$$

To obtain two orthogonal vectors which are orthogonal to $\boldsymbol{n}_c$, an auxiliary coordinate base ($\boldsymbol{n}_{\text{axis}}$) is built, which can be expressed as

$$\boldsymbol{n}_{\text{axis}} = \begin{cases} [1, 0, 0]^T, \text{ if } \min\left(|n_{cx}|, \left|n_{cy}\right|, |n_{cz}|\right) = |n_{cx}| \\ [0, 1, 0]^T, \text{ if } \min\left(|n_{cx}|, \left|n_{cy}\right|, |n_{cz}|\right) = \left|n_{cy}\right| \\ [0, 0, 1]^T, \text{ if } \min\left(|n_{cx}|, \left|n_{cy}\right|, |n_{cz}|\right) = |n_{cz}| \end{cases} \tag{12}$$

According to the geometric relationship between $\boldsymbol{n}_{axis}$, $\boldsymbol{n}_c$, and $\boldsymbol{n}_t$, the required two vector bases can be calculated as

$$\boldsymbol{b}_{c1} = \frac{\boldsymbol{n}_c \times \boldsymbol{n}_{\text{axis}}}{\|\boldsymbol{n}_c \times \boldsymbol{n}_{\text{axis}}\|} \tag{13}$$

$$\boldsymbol{b}_{c2} = \frac{\boldsymbol{n}_c \times \boldsymbol{b}_1}{\|\boldsymbol{n}_c \times \boldsymbol{b}_1\|} \tag{14}$$

It is easy to identified that $\boldsymbol{b}_{c1}$ is orthogonal to $\boldsymbol{b}_{c2}$. The difference between transformed normal vector and normal vector in the current frame $\boldsymbol{n}_t$-$\boldsymbol{n}_c$ is projected onto $\boldsymbol{b}_{c1}$ and $\boldsymbol{b}_{c2}$ to obtain the required a two-dimensional rotation residual, which can be calculated as

$$\boldsymbol{r}(\boldsymbol{R}_{wLc}, \boldsymbol{n}_w) = \boldsymbol{B}_c(\boldsymbol{n}_t - \boldsymbol{n}_c) = \boldsymbol{B}_c\left(\boldsymbol{R}_{wLc}{}^T\boldsymbol{n}_w - \boldsymbol{n}_c\right) \in \mathbb{R}^{2\times 1} \tag{15}$$

$$\boldsymbol{B}_c = \begin{bmatrix} \boldsymbol{b}_{c1}{}^T \\ \boldsymbol{b}_{c2}{}^T \end{bmatrix} \in \mathbb{R}^{2\times 3} \tag{16}$$

This rotation residual only affect the roll and pitch during optimization.

The position of LiDAR ($\boldsymbol{t}_{wLc}$) as well as the normal vector in the world map ($\boldsymbol{n}_w$) are employed to transfer the distance

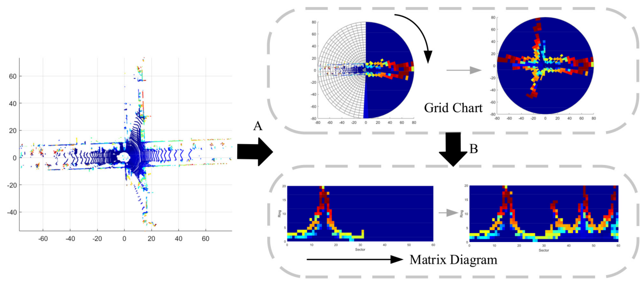


FIGURE 2. Schematic diagram of the loop closure descriptor.

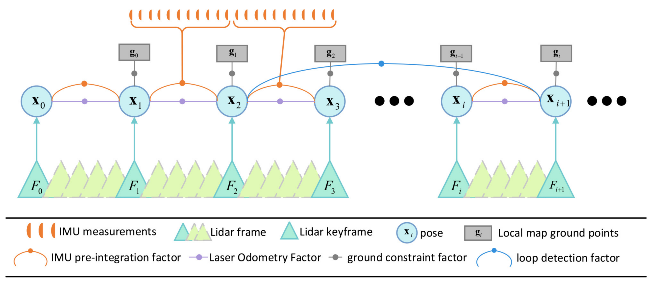


FIGURE 3. LiDAR-inertial back end factor graph model.

in the world map (d_w) to a transformed distance in the current frame (d_t), which can be calculated as

$$d_t = d_w - \boldsymbol{n}_w \boldsymbol{t}_{wLc} \tag{17}$$

The distance residual can be expressed as

$$d(\boldsymbol{t}_{wLc}, d_w) = d_t - d_c = d_w - d_c - \boldsymbol{n}_w \boldsymbol{t}_{wLc} \in \mathbb{R} \tag{18}$$

This distance residual only affects the position in the z-axis during optimization.

2) jacobian-calculation:

The residual of the ground constraint is denoted as

$$\boldsymbol{D}(\boldsymbol{T}_{wLc}, \boldsymbol{\Omega}_w) = \begin{bmatrix} \boldsymbol{r}(\boldsymbol{R}_{wLc}, \boldsymbol{n}_w) \\ d(\boldsymbol{t}_{wLc}, d_w) \end{bmatrix} \in \mathbb{R}^{3\times 1} \tag{19}$$

Derivative of residuals for each components are

$$\boldsymbol{J}^{2\times 3}_{\boldsymbol{R}_{wLc}} = \frac{\partial \boldsymbol{r}(\boldsymbol{R}_{wLc}, \boldsymbol{n}_w)}{\partial \boldsymbol{R}_{wLc}} = \boldsymbol{B}_t \boldsymbol{n}_t^{\wedge} \tag{20}$$

$$\boldsymbol{J}^{1\times 3}_{\boldsymbol{t}_{wLc}} = \frac{\partial d(\boldsymbol{t}_{wLc}, d_w)}{\partial \boldsymbol{t}_{wLc}} = -\boldsymbol{n}_w \tag{21}$$

$$\boldsymbol{J}^{3\times 3}_{\boldsymbol{\Omega}_w} = \frac{\partial \boldsymbol{D}(\boldsymbol{T}_{wLc}, \boldsymbol{\Omega}_w)}{\partial \boldsymbol{\Omega}_w} = \begin{bmatrix} \boldsymbol{B}_t \boldsymbol{R}_{wLc}{}^T \boldsymbol{B}_w{}^T & \boldsymbol{0} \\ \boldsymbol{t}_{wLc}{}^T \boldsymbol{B}_w{}^T & 1 \end{bmatrix} \tag{22}$$

where $\boldsymbol{J}^{2\times 3}_{\boldsymbol{R}_{wLc}}$, $\boldsymbol{J}^{1\times 3}_{\boldsymbol{t}_{wLc}}$, $\boldsymbol{J}^{3\times 3}_{\boldsymbol{\Omega}_w}$ are the derivatives of the residual to the rotation component, translation component, and world map ground point, respectively, and the upper right corner of the symbol indicates the dimension of the matrix.

2) LOOP CONSTRAINT FACTOR

Loop closure constraints are one of the most important constraints to eliminate cumulative errors in factor graph optimization. SC++ descriptor proposed by Giseop Kim [34] is used for environment re-identification, as shown in Fig. 2. Compared to the Scan Context [35] descriptor proposed in 2018, it not only increases lateral offset invariance, but also proposes two different forms of Polar Context (PC) descriptor relative to polar coordinates and Cart Context (CC) descriptor relative to Cartesian coordinates. The PC descriptor is selected for loop detection and constraint construction.

3) FACTOR GRAPH MODEL

The factor graph model of the LiDAR and IMU tightly coupled framework in this chapter is shown in Fig. 3, including IMU pre-integration factors, LiDAR odometry factors, loop detection factors, and ground constraint factors. Among them, the IMU factor is obtained by pre integrating the IMU measurement value between two consecutive key frames, the LiDAR odometry factor is obtained by matching the current frame point cloud with the world map point cloud, and the loop detection factor is obtained by establishing a loop between the current frame and the candidate frame. The ground constraint factors are obtained by constructing constraints between the current frame ground points and the world map ground points. These factors are added to the factor graph for global optimization to obtain the pose, using the optimized pose and the point cloud collected by the LiDAR, a global point cloud map can be obtained. The open-source library GTSAM is selected for optimization solution.

IV. EXPERIMENT VERIFICATIONS

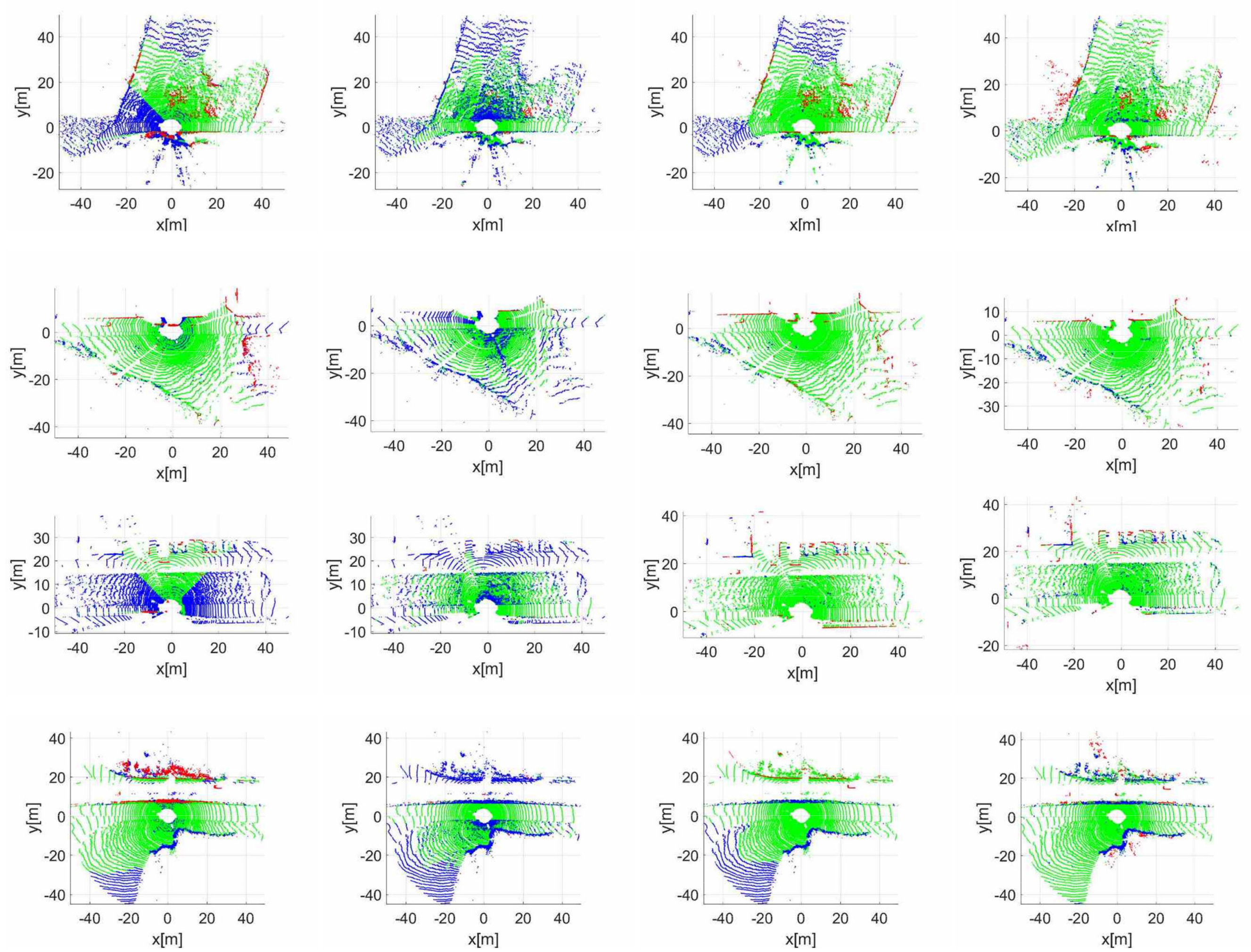
The proposed localization system is verified in two scenarios. First, the KITTI data is applied. Second, a real vehicle experiment in campus is employed.

A. KITTI DATASET VALIDATION

1) COMPARISON OF GROUND SEGMENTATION ALGORITHMS

The SemanticKITTI [36] dataset is selected as the ground truth, and the KITTI dataset is applied to run the GPF [37], LineFit [23], CascadedSeg [24] algorithms and the proposed ground segmentation algorithm (AdaptionBin) for analysis, of which the first three algorithms have open-source implementations.

Fig. 4 extracts multiple clips of the four algorithms running the KITTI dataset. The first line is the ground point segmentation of the 79th LiDAR key frame of the 10 sequence, the second line is the 439th LiDAR key frame of the 07 sequence, and the third line is the 226th LiDAR key frame of the 06 sequence, the fourth line is the 60th LiDAR key frame of the 01 sequence. Among them, the green points indicate the correctly segmented ground points, the blue points indicate that the actual ground points are determined as non-ground points by the algorithm, which is missed judgment, and the red points indicate that the actual



(a) CascadedSeg (b) LineFit (c) GPF (d) AdationBin

FIGURE 4. Ground segmentation effect display.

non-ground points are determined as ground points by the algorithm, which is a misjudgment.

It can be observed qualitatively from the Fig. 5 that compared with the proposed AdationBin algorithm, other algorithms have more missed judgments, which mainly affect the recall rate of the ground segmentation algorithm, while the AdationBin algorithm can better extract as much as possible multiple ground points.

The precision, recall and F1 scores of the four algorithms running multiple sequences in the KITTI dataset is quantitatively shown in Table 1. The precision, recall and F1 scores are defined as follows

$$\text{Precision} = \frac{N_{TP}}{N_{TP} + N_{FP}} \tag{23}$$

$$\text{Recall} = \frac{N_{TP}}{N_{TP} + N_{FN}} \tag{24}$$

$$\text{F1} = \frac{2N_{TP}}{2N_{TP} + N_{FP} + N_{FN}} \tag{25}$$

Among them, N_{TP} represents the number of points that are actually ground points and are detected by the algorithm as ground points, N_{FP} represents the number of points that are actually non-ground points but are detected by the algorithm as ground points, and N_{FN} represents the number of points that are actually ground points but are detected as non-ground points by the algorithm. Among them, the precision rate can reflect the false detection rate of the algorithm, the recall rate can reflect the missed detection rate of the algorithm, and the F1 score is a combination of these two indicators.

The dataset sequences selected for comparison include urban roads (U), highways (H), rural roads (C), which can more fully reflect the advantages and disadvantages of various algorithms. The part marked in red in the table indicates that the algorithm performs best under this indicator, and the part marked in blue indicates that the algorithm performs second best under this indicator. From the analysis of the accuracy, it can be seen that the LineFit algorithm and the proposed AdationBin algorithm occupy the optimal and suboptimal positions respectively, and the LineFit algorithm is slightly better. From the analysis of the recall rate, it can be concluded that the GPF algorithm and the AdationBin algorithm perform best

TABLE 1. The indicators of the four algorithms running the KITTI dataset.

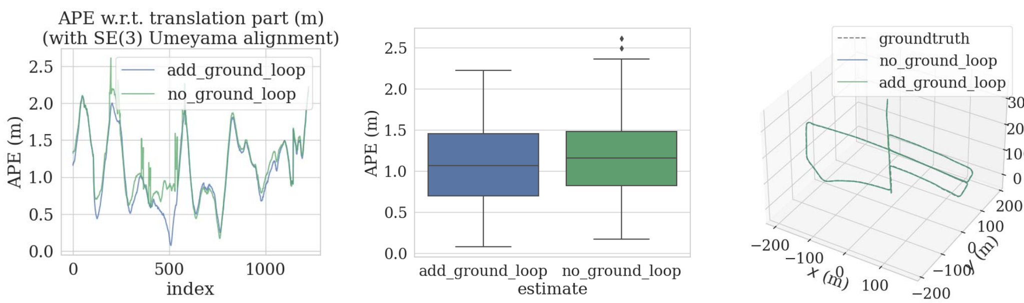
Evaluation	Method	KITTI dataset						
		00U	01H	06U	07U	10C	Mean	Covariance
Precision	CascadedSeg	69.91	93.60	89.83	65.74	57.39	80.01	19.85
	Linefit	98.15	98.89	99.23	97.15	96.90	97.85	1.66
	GPF	91.77	97.27	94.44	89.75	87.35	92.71	4.88
	AdationBin	92.72	95.91	96.67	90.21	89.20	93.04	4.17
Recall	CascadedSeg	67.85	82.06	81.03	67.05	56.39	74.18	16.10
	Linefit	83.05	74.22	81.23	90.05	78.05	81.46	8.87
	GPF	94.46	95.12	97.96	97.32	87.60	93.91	5.55
	AdationBin	94.37	89.13	97.04	96.73	90.32	93.66	4.33
F1 score	CascadedSeg	68.86	87.45	85.21	66.39	56.89	76.98	17.48
	Linefit	89.97	84.79	89.34	93.47	86.46	88.91	5.61
	GPF	93.10	96.18	96.17	93.38	87.48	93.31	3.71
	AdationBin	93.54	92.39	96.85	93.36	89.76	93.35	3.14

TABLE 2. Algorithm time-consuming.

Strategy	Baseline	Algorithm	Time-consuming /ms
Line fitting	-	LineFit	75
Plane fitting	RANSAC	CascadedSeg	83
	PCA	GPF	17
		AdationBin	16

TABLE 3. Absolute trajectory error parameter.

Sequence	Loop	Ground constraint	Max	Min	Std	Median	Mean	RMSE	Accuracy improvement
05	no	add	7.88	0.75	1.31	1.71	2.31	2.66	13.75%
		no	8.69	1.19	1.55	2.11	2.66	3.08	
	add	add	2.22	0.07	0.47	1.06	1.09	1.19	7.63%
		no	2.60	0.17	0.45	1.15	1.18	1.27	
07	no	add	0.95	0.09	0.21	0.38	0.43	0.48	8.51%
		no	1.18	0.07	0.19	0.43	0.47	0.50	
	add	add	0.81	0.07	0.17	0.38	0.39	0.43	7.14%
		no	0.82	0.11	0.18	0.37	0.42	0.46	
09	no	add	19.56	0.59	5.76	5.94	7.69	9.61	3.03%
		no	20.49	0.51	6.31	6.32	7.93	10.13	
	add	add	2.75	0.47	0.54	1.33	1.40	1.51	4.76%
		no	4.28	0.52	0.71	1.26	1.47	1.63	



(a) Absolute translation error graph (b) Box plot (c) Trajectory comparison chart

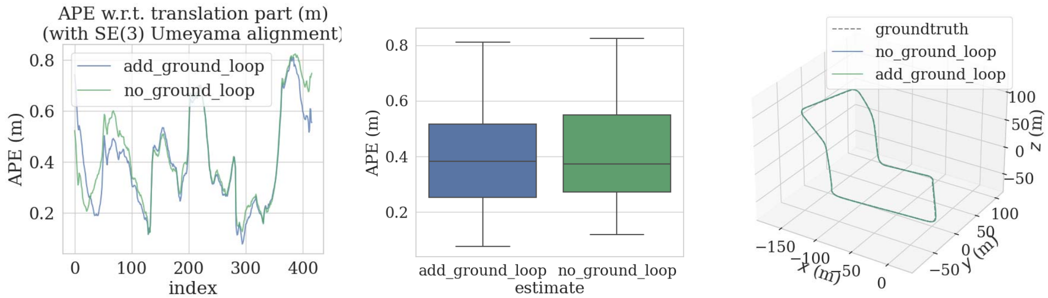
FIGURE 5. KITTI_05 sequence absolute error and trajectory comparison chart.

respectively. According to the comprehensive analysis of the two indicators according to the F1 score, the AdationBin algorithm performs best.

2) ANALYSIS OF LOCALIZATION RESULTS

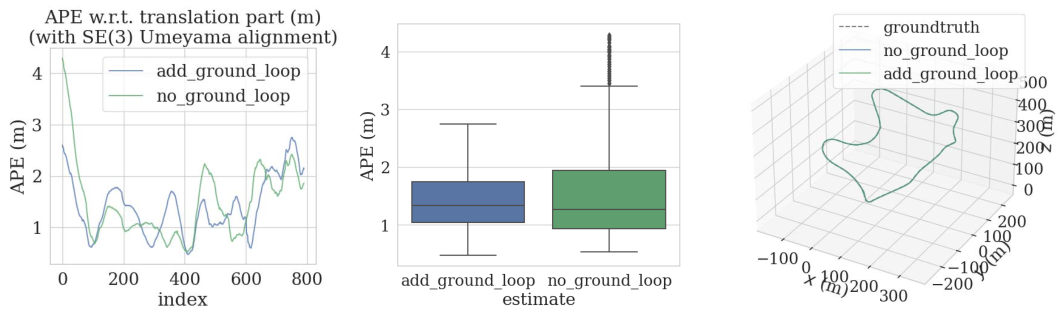
According to the analysis of the real-time performance of the four algorithms in Table 2, the PCA-based algorithm runs faster than the RANSAC-based algorithm, and the plane fitting strategy has a lower calculation load than the straight line fitting strategy. Among them, although the LineFit algorithm has higher accuracy than the AdationBin algorithm, the calculation efficiency is very low, so the proposed AdationBin algorithm has the best performance in terms of ground segmentation and calculation efficiency.

According to the trajectory comparison in Fig. 5, Fig. 6, and Fig. 7, it can be seen that the result of LiDAR and IMU tightly coupled SLAM algorithm proposed in this paper is closer to the true value in different environments. In order to compare the impact of ground constraints on positioning accuracy, ablation experiments for whether ground constraints are added to the



(a) Absolute translation error graph (b) Box plot (c) Trajectory comparison chart

FIGURE 6. KITTI_07 sequence absolute error and trajectory comparison chart.



(a) Absolute translation error graph (b) Box plot (c) Trajectory comparison chart

FIGURE 7. KITTI_09 sequence absolute error and trajectory comparison chart.

TABLE 4. Quantitative analysis under test site conditions (Unit: m).

Scenario	Algorithm	Max	Min	Std	Median	Mean	RMSE
Open area	GLI-SLAM	5.93	0.07	0.58	1.00	1.01	1.17
	LeGO-LOAM	6.18	0.06	0.57	1.01	1.02	1.17
Garden area	GLI-SLAM	2.99	0.55	0.39	1.63	1.61	1.66
	LeGO-LOAM	3.01	0.67	0.38	1.67	1.65	1.69

SLAM algorithm are designed, and the absolute translation error of the positioning results are calculated. Considering that the promotion effect of ground constraints on positioning is mainly reflected in relatively flat road surfaces, the SLAM algorithm only adds ground constraints on relatively gentle road sections, and the experimental results are shown in Fig. 5~7(a) and 5~7(b). According to the curve diagram and box diagram, it can be seen that adding ground constraints on relatively gentle road sections can improve the positioning accuracy of the proposed SLAM algorithm, and the robustness is better.

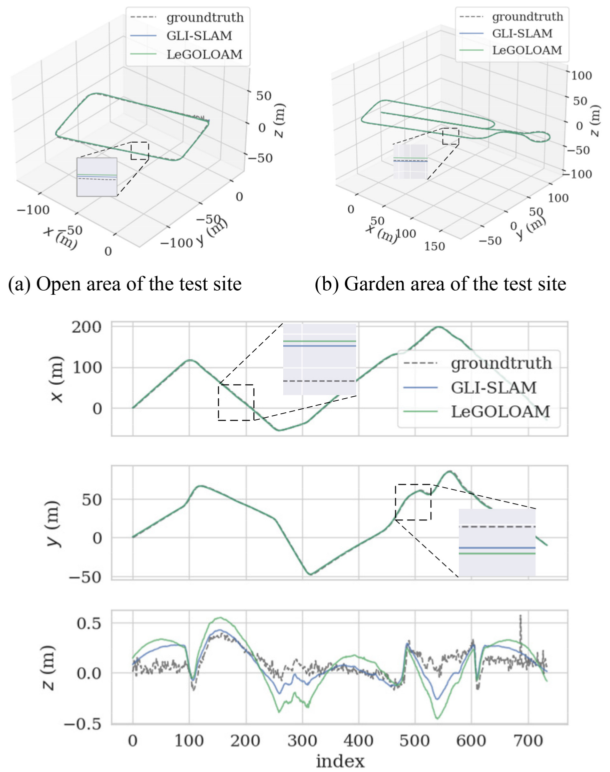
The impact of ground constraints on SLAM positioning accuracy can be quantitatively analyzed according to Table 3. From the error results of the 05 and 07 sequences, it can be seen that adding ground constraints significantly improves the accuracy of urban environment positioning. Additionally, since the distance between the 05 dataset and the 07 dataset is greater, it can be analyzed that the longer the running trajectory, the more accuracy improvement can be achieved by adding ground constraints, provided that there are more flat road surfaces. As a rural road, the 09 dataset has significant elevation changes, and adding ground constraints has a small impact on accuracy improvement.

FIGURE 8. Data collection platform.

(a) Open area of the test site (b) Garden area of the test site

FIGURE 9. Satellite map of the test site.

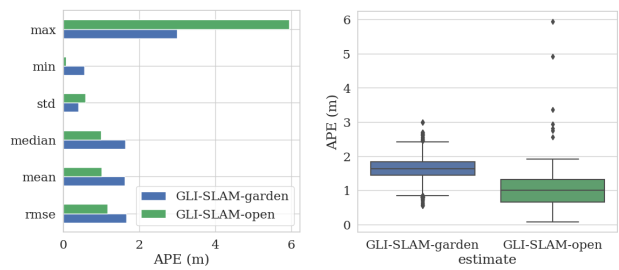
For the same dataset with and without loop detection, analyzing whether to add ground constraints has an impact on positioning accuracy. In the 05 and 07 sequences, adding ground constraints with loop detection has a smaller



(a) Open area of the test site (b) Garden area of the test site

(c) Comparison of three axis errors in garden area of the test site

FIGURE 10. Qualitative analysis of GLI-SLAM and LeGO-LOAM.



(a) (b)

FIGURE 11. Quantitative analysis and statistics of automobile test field.

improvement in accuracy compared to not adding loop detection. This article analyzes that this is because the introduction of loop detection can eliminate some of the errors caused by not adding ground constraints, Thus, adding ground constraints reduces the improvement in accuracy.

B. CAMPUS DATA VERIFICATION

The data collection platform is built based on LYNK&CO 02, as shown in Fig. 8. The sensors of the smart car platform mainly include 64-line LiDAR, binocular camera, AHRS, and RTK, and an automatic driving industrial computer is used for data storage and processing.

A vehicle test field is selected to verify the real vehicle data of the localization and mapping algorithm. Two typical environments are selected. One is the open area of the test site, with a total length of about 501m, and the other is the flower bed area of the test site, with a total length of about 942m. The former has less environmental information, and the point clouds collected by LiDAR are mostly ground points. Fig. 9 is a satellite image of the vehicle test site, where the red line is the trajectory projected from the RTK output data.

During data collection, the experimental vehicle maintains a speed of 20km/h, and slows down appropriately during the turning process. Finally, the collected data (rosbag package) will be played at the same frequency as when it was collected, so as to run the proposed algorithm called GLI-SLAM and the open-source LeGO-LOAM algorithm respectively, and compare the pose output results. The output trajectory is shown in Fig. 10.

According to the comparative trajectories of the two algorithms (a) and (b), it can be seen that GLI-SLAM and LEGO-LOAM can achieve stable pose output in both open areas and flower beds, with a small difference from the ground truth of RTK. From the locally enlarged image, it can be seen that the trajectory output of the proposed GLI-SLAM is closer to the ground truth. Analyzing the error comparison diagram of the *xyz* coordinate in the flower bed area (c), it can be concluded that GLI-SLAM has a significant advantage in reducing the z-axis or elevation error compared to LeGO-LOAM. It is analyzed that this is the ground constraint module in factor graph optimization that plays a role.

The absolute trajectory error (APE) is shown in Table 4. It is shown that the accuracy of the GLI-SLAM algorithm and the LeGO-LOAM algorithm under different operating conditions was quantitatively analyzed. After calculation, compared to the LeGO-LOAM algorithm, the mean square error of GLI-SLAM in the open area decreased by about 1.75%, and the mean square error in the flower bed area decreased by about 2.63%. Compared to open areas, the accuracy improvement effect of GLI-SLAM in the flower bed area is more significant. From this, it can be seen that the more complex the environment and the longer the running time are, the more significant the error reduction effect of GLI-SLAM is, and the more it can reflect the advantages of the algorithm proposed in this article.

Fig 11. shows a visual statistics of Table 4, where garden represents the garden area and open represents the open area. Since there is a certain jump in solving the RTK pose, the maximum error is not considered, and it is regarded as outlier. By using error statistics (a) and box plots (b), it can be concluded that GLI-SLAM performs better in open areas.

V. CONCLUSION

A LiDAR-inertial localization system based on ground constraints is proposed to resolve the inaccurate height positioning. In order to obtain ground points with higher verticality, the ground segmentation algorithm Patchwork is improved and is added to the LiDAR-inertial odometry. It constructs constraints using the current frame ground points and the world map ground points, and adds them to the factor graph optimization to limit elevation errors. At the same time, advanced SC++descriptors are used to construct loop constraints to eliminate cumulative errors. Through the validation of the KITTI dataset, it can be concluded that the proposed ground segmentation algorithm has high accuracy, and through ablation experiments, it is found that ground constraints can effectively limit the elevation error of the localization system. Finally, by building an intelligent vehicle data acquisition platform for real vehicle data validation, compared with existing algorithms, the proposed localization algorithm has higher accuracy.

REFERENCES

[1] J. Betz et al., "Autonomous vehicles on the edge: A survey on autonomous vehicle racing," *IEEE Open J. Intell. Transp. Syst.*, vol. 3, pp. 458–488, 2022.

[2] C. Plachetka, B. Sertolli, and J. Fricke, "DNN-based map deviation detection in LiDAR point clouds," *IEEE Open J. Intell. Transp. Syst.*, vol. 4, pp. 580–601, 2023.

[3] R. Trauth, K. Moller, and J. Betz, "Toward safer autonomous vehicles: Occlusion-aware trajectory planning to minimize risky behavior," *IEEE Open J. Intell. Transp. Syst.*, vol. 4, pp. 929–942, 2023.

[4] D. P. Bavirisetti, H. R. Martinsen, G. H. Kiss, and F. Lindseth, "A multi-task vision transformer for segmentation and monocular depth estimation for autonomous vehicles," *IEEE Open J. Intell. Transp. Syst.*, vol. 4, pp. 909–928, 2023.

[5] M. Sakthi, M. Arvinte, and H. Vikalo, "Automotive radar sub-sampling via object detection networks: Leveraging prior signal information," *IEEE Open J. Intell. Transp. Syst.*, vol. 4, pp. 858–869, 2023.

[6] J. Lee, G. Bang, T. Nishimori, K. Nakao, and S. Kamijo, "Radar translation network between sunny and rainy domains by combination of KP-convolution and CycleGAN," *IEEE Open J. Intell. Transp. Syst.*, vol. 4, pp. 833–845, 2023.

[7] Y. Qian, X. Wang, H. Zhuang, C. Wang, and M. Yang, "3-D vehicle detection enhancement using tracking feedback in sparse point clouds environments," *IEEE Open J. Intell. Transp. Syst.*, vol. 4, pp. 471–480, 2023.

[8] Z. Wang, P. Keo, and M. Saberi, "Real-time traffic state measurement using autonomous vehicles open data," *IEEE Open J. Intell. Transp. Syst.*, vol. 4, pp. 602–610, 2023.

[9] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardós, "ORB-SLAM: A versatile and accurate monocular SLAM system," *IEEE Trans. Robot.*, vol. 31, no. 5, pp. 1147–1163, Oct. 2015.

[10] R. Mur-Artal and J. D. Tardós, "ORB-SLAM2: An open-source slam system for monocular, stereo, and RGB-D cameras," *IEEE Trans. Robot.*, vol. 33, no. 5, pp. 1255–1262, Oct. 2017.

[11] T. Qin, P. Li, and S. Shen, "VINS-mono: A robust and versatile monocular visual-inertial state estimator," *IEEE Trans. Robot.*, vol. 34, no. 4, pp. 1004–1020, Aug. 2018.

[12] C.-L. Lee, C.-Y. Hou, C.-C. Wang, and W.-C. Lin "Extrinsic and temporal calibration of automotive radar and 3-D LiDAR in factory and on-road calibration settings," *IEEE Open J. Intell. Transp. Syst.*, vol. 4, pp. 708–719, 2023.

[13] J. Zhang and S. S. Singh, "LOAM: LiDAR odometry and mapping in real-time," *Robot., Sci. Syst.*, vol. 2, no. 9, pp. 1–9, 2014.

[14] H. Ye, Y. Chen, and M. Liu, "Tightly coupled 3D LiDAR inertial odometry and mapping," in *Proc. Int. Conf. Robot. Autom. (ICRA)*, 2019, pp. 3144–3150.

[15] D. Droeschel and S. Behnke, "Efficient continuous-time SLAM for 3D LiDAR-based online mapping," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, 2018, pp. 5000–5007.

[16] F. Neuhaus, T. Koß, R. Kohnen, and D. Paulus, "MC2SLAM: Real-time inertial LiDAR odometry using two-scan motion compensation," in *Proc. German Conf. Pattern Recognit.*, 2018, pp. 60–72.

[17] J. Laconte, S.-P. Deschênes, M. Labussière, and F. Pomerleau, "LiDAR measurement bias estimation via return waveform modelling in a context of 3D mapping," in *Proc. Int. Conf. Robot. Autom. (ICRA)*, 2019, pp. 8100–8106.

[18] B. Schlager, T. Goelles, S. Muckenhuber, and D. Watzenig, "Contaminations on LiDAR sensor covers: Performance degradation including fault detection and modeling as potential applications," *IEEE Open J. Intell. Transp. Syst.*, vol. 3, pp. 738–747, 2022.

[19] B. Schlager, T. Goelles, M. Behmer, S. Muckenhuber, J. Payer, and D. Watzenig, "Automotive LiDAR and vibration: Resonance, inertial measurement unit, and effects on the point cloud," *IEEE Open J. Intell. Transp. Syst.*, vol. 3, pp. 426–434, 2022.

[20] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? The KITTI vision benchmark suite," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2012, pp. 3354–3361.

[21] K. Na, B. Park, and B. Seo, "Drivable space expansion from the ground base for complex structured roads," in *Proc. IEEE Int. Conf. Syst., Man, Cybern. (SMC)*, 2016, pp. 000373–000378.

[22] J. Byun, K.-i. Na, B.-s. Seo, and M. Roh, "Drivable road detection with 3D point clouds based on the MRF for intelligent vehicle," in *Field and Service Robotics: Results of the 9th International Conference*. Cham, Switzerland: Springer, 2015.

[23] M. Himmelsbach, F. v. Hundelshausen, and H.-J. Wuensche, "Fast segmentation of 3D point clouds for ground vehicles," in *Proc. IEEE Intell. Veh. Symp.*, 2010, pp. 560–565.

[24] P. Narksri, E. Takeuchi, Y. Ninomiya, Y. Morales, N. Akai, and N. Kawaguchi, "A slope-robust cascaded ground segmentation in 3D point cloud for autonomous vehicles," in *Proc. Int. Conf. Intell. Transp. Syst. (ITSC)*, 2018, pp. 497–504.

[25] H. Lim, S. Hwang, and H. Myung, "ERASOR: Egocentric ratio of pseudo occupancy-based dynamic object removal for static 3D point cloud map building," *IEEE Robot. Autom. Lett.*, vol. 6, no. 2, pp. 2272–2279, Apr. 2021.

[26] H. Lim, M. Oh, and H. Myung, "Patchwork: Concentric zone-based region-wise ground segmentation with ground likelihood estimation using a 3D LiDAR sensor," *IEEE Robot. Autom. Lett.*, vol. 6, no. 4, pp. 6458–6465, Oct. 2021.

[27] M. Kaess, "Simultaneous localization and mapping with infinite planes," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, 2015, pp. 4605–4611.

[28] T. Shan and B. Englot, "LeGO-LOAM: Lightweight and ground-optimized LiDAR odometry and mapping on variable terrain," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, 2018, pp. 4758–4765.

[29] X. Liu et al., "Optimized LOAM using ground plane constraints and SegMatch-based loop detection," *Sensors*, vol. 19, no. 23, p. 5419, 2019.

[30] X. Wei et al., "Ground-SLAM: Ground constrained LiDAR SLAM for structured multi-floor environments," 2021, *arXiv:2103.03713*.

[31] Y. H. Khalil and H. T. Mouftah, "LiCaNet: Further enhancement of joint perception and motion prediction based on multi-modal fusion," *IEEE Open J. Intell. Transp. Syst.*, vol. 3, pp. 222–235, 2022.

[32] S. Gumhold, X. Wang, and R. MacLeod, "Feature extraction from point clouds," 2001, submitted for publication.

[33] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[34] G. Kim, S. Choi, and A. Kim, "Scan context++: Structural place recognition robust to rotation and lateral variations in urban environments," *IEEE Trans. Robot.*, vol. 38, no. 3, pp. 1856–1874, Jun. 2022.

[35] G. Kim and A. Kim, "Scan context: Egocentric spatial descriptor for place recognition within 3D point cloud map," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, 2018, pp. 4802–4809.

[36] J. Behley et al., "SemanticKITTI: A dataset for semantic scene understanding of LiDAR sequences," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 9296–9306.

[37] D. Zermas, I. Izzat, and N. Papanikolopoulos, "Fast segmentation of 3D point clouds: A paradigm on LiDAR data for autonomous vehicle applications," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, 2017, pp. 5067–5073.

MAN YU received the M.S. and Ph.D. degrees in vehicle engineering from Chang'an University, Xi'an, China, in 2013 and 2018, respectively, where she is currently a Senior Engineer with the School of Construction Machinery. Her research interests include intelligent transportation systems and electric vehicle control.

KEYANG GONG received the B.S. and M.S. degrees in vehicle engineering from Chang'an University, Xi'an, China, in 2019 and 2023, respectively. He is currently an Engineer with the Intelligent Driving Center, Geely Automotive Research Institute (Ningbo) Co. His research interests including SLAM and automated parking systems.

WEIHUA ZHAO received the B.S. and M.S. degrees in vehicle operation engineering and the Ph.D. degree in transportation engineering from Chang'an University, Xi'an, China, in 2000, 2004, and 2010, respectively. He is currently a Professor with the School of Vehicle Engineering, Xi'an Aeronautical Institute. His research interests include intelligent traffic safety and special vehicle structure.

RUI LIU received the B.S. and M.S. degrees in vehicle engineering from Chang'an University, Xi'an, China, in 2012 and 2015, respectively, and the Ph.D. degree in vehicle engineering from Tongji University, Shanghai, China, in 2020. He is currently an Associate Professor with the School of Automobile, Chang'an University. His research interests include intelligent vehicle perception, decision, and evaluation.