remote sensing

MDPI

Article

GNSS/INS/LiDAR-SLAM Integrated Navigation System Based on Graph Optimization

Le Chang, Xiaoji Niu *, Tianyi Liu, Jian Tang and Chuang Qian

GNSS Research Center, Wuhan University, 129 Luoyu Road, Wuhan 430079, China;
changlesgg@whu.edu.cn (L.C.); liutianyi@whu.edu.cn (T.L.); tangjian@whu.edu.cn (J.T.);
qc_gnss@whu.edu.cn (C.Q.)
* Correspondence: xjniu@whu.edu.cn; Tel.: +86-186-0270-8312

Received: 3 April 2019; Accepted: 23 April 2019; Published: 28 April 2019

Abstract: A Global Navigation Satellite System (GNSS)/Inertial Navigation System (INS)/Light Detection and Ranging (LiDAR)-Simultaneous Localization and Mapping (SLAM) integrated navigation system based on graph optimization is proposed and implemented in this paper. The navigation results are obtained by the information fusion of the GNSS position, Inertial Measurement Unit (IMU) preintegration result and the relative pose from the 3D probability map matching with graph optimizing. The sliding window method was adopted to ensure that the computational load of the graph optimization does not increase with time. Land vehicle tests were conducted, and the results show that the proposed GNSS/INS/LiDAR-SLAM integrated navigation system can effectively improve the navigation positioning accuracy compared to GNSS/INS and other current GNSS/INS/LiDAR methods. During the simulation of one-minute periods of GNSS outages, compared to the GNSS/INS integrated navigation system, the root mean square (RMS) of the position errors in the North and East directions of the proposed navigation system are reduced by approximately 82.2% and 79.6%, respectively, and the position error in the vertical direction and attitude errors are equivalent. Compared to the benchmark method of GNSS/INS/LiDAR-Google Cartographer, the RMS of the position errors in the North, East and vertical directions decrease by approximately 66.2%, 63.1% and 75.1%, respectively, and the RMS of the roll, pitch and yaw errors are reduced by approximately 89.5%, 92.9% and 88.5%, respectively. Furthermore, the relative position error during the GNSS outage periods is reduced to 0.26% of the travel distance for the proposed method. Therefore, the GNSS/INS/LiDAR-SLAM integrated navigation system proposed in this paper can effectively fuse the information of GNSS, IMU and LiDAR and can significantly mitigate the navigation error, especially for cases of GNSS signal attenuation or interruption.

Keywords: graph optimization; GNSS/INS; LiDAR-SLAM; integrated navigation; IMU preintegration

1. Introduction

With the rapid development of autonomous driving and intelligent robots, the demand for navigation information with high data rates, high precision and all-weather features continues to increase, especially in complex urban environments. Thus, a single navigation technique can hardly meet the requirements for practical applications, and the synthesis of multiple navigation techniques has become the development trend in navigation. Among the various synthesized navigation techniques, the Global Navigation Satellite System/Inertial Navigation System (GNSS/INS) integrated navigation system, which is dominated by the INS and supplemented by the GNSS, is the most popular. The characteristics of the GNSS and the INS are distinctively complementary: (1) generally, information from GNSS and INS are effectively integrated via Kalman filtering [1]; and (2) INS largely compensates for the shortcoming of GNSS, which is vulnerable to interference, while GNSS provides the periodic correction information for INS; hence, the synergy of the two techniques improves the navigation performance. However, in areas where the GNSS signal is not available, the GNSS/INS integrated navigation system relies on the performance of INS alone and experiences drifting errors. In addition, for the low-cost MEMS (Micro-Electro-Mechanical System)-Inertial Measurement Unit (IMU), the navigation errors will accumulate and increase rapidly over time. In the context of low-cost integrated systems, image-based methodologies have also been explored, aiming at investigating the impact of drifts and errors experienced with such techniques, like in [2,3]. In the field of computer vision and robotics, Simultaneous Localization and Mapping (SLAM) technology is widely used for navigation in unfamiliar environments. The SLAM system locates itself mainly according to position estimation and a map during movement and builds up incremental maps based on the self-localization to achieve independent positioning and navigation [4]. In recent years many excellent SLAM systems based on a single sensor have been developed, such as LG-SLAM [5], IMLS-SLAM [6], SUMA [7] and LOAM [8]. VINS [9] uses a tightly coupled, nonlinear optimization-based method to obtain highly accurate visual-inertial odometry by fusing preintegrated IMU measurements and feature observations. vLOAM [10] presents a general framework for combining camera and Light Detection and Ranging (LiDAR). However, the errors of these SLAM systems, which are reduced mainly by the closed-loop correction method, will also increase with the moving distance. In large-scale outdoor motion, SLAM is less likely to form a closed-loop. Additionally, SLAM cannot provide absolute navigation information. Therefore, the combination of the GNSS/INS integrated navigation system and SLAM will effectively mitigate the navigation drift when the GNSS signal is not available, reduce the dependence of SLAM on closed-loop correction and provide absolute navigation information to fulfill the complementary advantages of the three techniques.

Generally, cameras and LiDAR are the two most common sensors for SLAM, and each of the two types of sensors has strengths and weaknesses. Compared to the camera, LiDAR, despite a higher cost and lower resolution, can directly obtain the structure information of environments and is hardly influenced by light or weather. Therefore, in this paper, mechanical rotating LiDAR is adopted.

In order to estimate pose change from LiDAR measurements, there are about three different categories of scan matching method: feature-based scan matching, point-based scan matching and mathematical property-based scan matching [11]. The feature-based scan matching is matching with some key elements which can be geometric primitives such as points, lines, and polygons, or a combination thereof in the LiDAR data. The point-based scan matching directly searches and matches the corresponding points in the LiDAR data. The Iterative Closest Point (ICP) algorithm and its variants [12,13] are the most popular methods to solve the point-based scan matching problem. The mathematical property-based scan matching can use various mathematical properties, such as the Normal Distribution Transform (NDT) [14] or the probability grid [15] to depict scan data changes. The feature-based scan matching method is efficient and accurate. But, it relies on features extracted from the environment. It may fail to work properly in outdoor or indoor unconstructed environments. The point-based scan matching method is accurate and independent of environment features. But, it takes a long time because of the inevitable iteration [16]. Thereby in this paper, the probability grid scan matching is used for the point cloud matching in the outdoor environment.

For data fusion in SLAM, there are mainly two methods: filtering and graph optimization. The former is better than the latter in terms of calculation speed but inferior in accuracy [17]. Qian [18] and Gao [16] used the EKF (extended Kalman filter) to perform a combination of GNSS, INS and LiDAR-SLAM. Shamsudin [19] used particle filtering to combine GNSS with LiDAR-SLAM. There has been increasing research on graph optimization in recent years. Kukko [20] used the graph optimization method to combine the results of the GNSS/INS with a single-line LiDAR. Hess [15] implemented 2D-LiDAR navigation based on graph optimization. Pierzchała [21] used graph optimized 3D LiDAR-SLAM for woods construction. Cartographer [22] completed a fusion of GNSS, 3D-LiDAR and IMU based on graph optimization. However, it was assumed that the vehicle was moving at a low speed and moved smoothly. Therefore, gravity is used to solve the horizontal attitude (i.e., roll and pitch), and error modelling for the IMU bias was ignored; the estimation of the vehicle's velocity was also ignored.

Considering the limitations of Cartographer and based on the Cartographer codes, a GNSS/INS/LiDAR-SLAM integrated navigation system is implemented in this paper based on graph optimization. First, the MEMS-IMU mechanization is applied to predict the motion of the vehicle and provide the searching initial value for probability map scan matching with LiDAR in the front-end. In the back-end, the GNSS position provides the absolute position constraint, while the IMU preintegration [9,23] is applied to increase the motion constraints, and the LiDAR-SLAM scan matching provides the relative pose constraints. Then, three information sources are combined by graph optimization to obtain the final navigation positioning results. In addition, to keep the parameters of the graph optimization invariant with time, a sliding window is applied to ensure the relative stability of the graph optimization parameter number.

The remainder of the paper is organized as follows. In Section 2, the mathematical model of the GNSS/INS/LiDAR-SLAM integrated navigation system is described. In Section 3, the land vehicle tests are introduced. The experimental results are discussed in Section 4. Finally, in Section 5, conclusions and future prospects are presented.

2. GNSS/INS/LiDAR-SLAM Integrated Navigation System

The GNSS/INS/LiDAR-SLAM integrated navigation system, an overview of which is shown in Figure 1, mainly is comprised of two parts: scan matching in the front-end and graph optimization in the back-end. In the front-end scan matching process, the point cloud acquired by LiDAR is first filtered by a voxel filter. Then, with the MEMS-IMU mechanization result, a node is formed, and the initial search value appears. Subsequently, probabilistic map scan matching is conducted for the node, and the node coordinate is obtained in the local coordinate system. Then, the node that meets certain conditions is selected to be inserted into the probability submap, and finally, the probability submap is updated. In the back-end graph optimization process, the cost functions are constructed with the GNSS position observation, the result of the IMU preintegration between two adjacent nodes and the relative pose constraints between the nodes and submaps. A sliding window is applied to ensure the relative stability of the number of optimized variables.

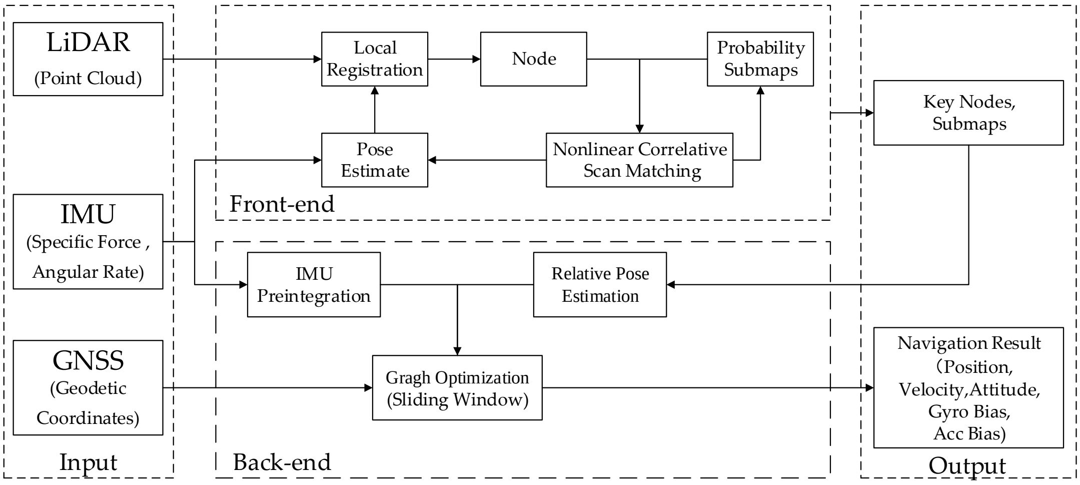

Figure 1. System overview of Global Navigation Satellite System (GNSS)/Inertial Navigation System (INS)/Light Detection and Ranging (LiDAR)-Simultaneous Localization and Mapping (SLAM) integrated navigation.

2.1. Coordinate Frame

The GNSS/INS/LiDAR-SLAM integrated navigation system involves multiple coordinate systems, and information fusion in different coordinate systems is needed. The coordinate systems and transformation formula used in the GNSS/INS/LiDAR-SLAM integrated navigation system are given below.

2.1.1. Coordinate System

1. Inertial Coordinate System (i-frame)

The inertial coordinate system is a coordinate system that is invariant in space. The Earth's inertial coordinate system is usually used to study the motion of the vehicles near the Earth's surface.

2. Earth-Centered Earth-Fixed Coordinate System (e-frame)

The Earth-centered Earth-fixed coordinate system rotates with the Earth, taking the Earth's centroid as the coordinate origin, and the X axis points to the intersection of the equator and the prime meridian. The Earth's rotation axis is taken as the Z axis, and the North Pole is the positive direction. Then, the Y axis is perpendicular to the X-Z plane, forming a right-handed coordinate system.

3. World Coordinate System (w-frame)

The world coordinate system is used to express the GNSS positioning results. The origin is the initial GNSS position, the X-Y plane is the local horizontal plane, with the X axis being the tangential line of the latitude line that points east, and the Y axis is the tangential line of the longitude line that points North. The Z axis is perpendicular to the X-Y plane, forming a right-handed coordinate system.

4. IMU Coordinate System (b-frame)

The IMU coordinate system moves with the vehicle, with the IMU barycenter as the coordinate origin, the X axis points to the right along the IMU horizontal axis, the Y axis points forward along the IMU longitudinal axis, and the Z axis is perpendicular to the X-Y plane, forming a right-handed coordinate system.

5. LiDAR Coordinate System (l-frame)

The LiDAR coordinate system moves with the vehicle, with the LiDAR measurement center as the coordinate origin, the X axis pointing to the right along the LiDAR horizontal axis, the Y axis pointing forward along the LiDAR longitudinal axis, and the Z axis being perpendicular to the X-Y plane, forming a right-handed coordinate system.

6. Map Coordinate System (m-frame)

The map coordinate system is used by LiDAR-SLAM. In this frame, the origin is located where the SLAM is initialized, the X-Y plane is the local horizontal plane, the direction of the X axis is indeterminate, and the yaw between the b-frame and m-frame is 0° on initialization.

2.1.2. Transformation Among Coordinate Systems

All the coordinate transformations in this paper are rigid body transformations. The rigid body transformation from the a-frame to b-frame is defined as follows:

$$\mathbf{R}_a^b = \left(\mathbf{p}_a^b, \mathbf{q}_a^b\right) \tag{1}$$

where $\mathbf{p}_a^b$ and $\mathbf{q}_a^b$ are the translation and quaternion from the a-frame to b-frame, respectively.

The rigid body transformation operation is defined as follows:

$$\begin{aligned} \mathbf{R}_b^a &= \left(\mathbf{R}_a^b\right)^{-1} = \left(-\left(\mathbf{C}_a^b\right)^T \cdot \mathbf{p}_a^b, \left(\mathbf{q}_a^b\right)^{-1}\right) \\ \mathbf{v}^b &= \mathbf{R}_a^b \mathbf{v}^a = \mathbf{C}_a^b \mathbf{v}^a + \mathbf{p}_a^b \\ \mathbf{R}_c^a &= \mathbf{R}_b^a \mathbf{R}_c^b = \left(\mathbf{C}_b^a \mathbf{p}_c^b + \mathbf{p}_b^a, \mathbf{q}_b^a \otimes \mathbf{q}_c^b\right) \end{aligned} \tag{2}$$

where $\mathbf{C}_a^b$ is the direction cosine matrix of $\mathbf{q}_a^b$ and $\mathbf{v}^a$ is a vector in the a-frame.

2.2. Front-End Scan Matching

2.2.1. Pose Estimate

In the m-frame, the pose estimate starts from the origin, and the initial yaw between the b-frame and m-frame is set to 0°. The roll and pitch are initialized by gravity when the vehicle is at rest at the beginning. Then, the velocity, position and attitude are updated by the following MEMS-IMU mechanization.

1. Velocity Update

The IMU measures the angular rate $\tilde{\boldsymbol{\omega}}_{ib}^{b}$ between the b-frame and i-frame and the specific force $\tilde{\mathbf{f}}^{b}$. The velocity differential equation of the strapdown inertial navigation in the e-frame can be obtained by the Coriolis's theorem [24]:

$$\dot{\mathbf{v}}^{e} = \mathbf{C}_{b}^{e}\mathbf{f}^{b} + \mathbf{g}^{e} - 2\boldsymbol{\omega}_{ie}^{e} \times \mathbf{v}^{e} \tag{3}$$

where $\mathbf{v}^{e}$ is the velocity in the e-frame; $\mathbf{C}_{b}^{e}$ is the direction cosine matrix of the attitude from the b-frame to the e-frame; $\mathbf{g}^{e}$ is gravity in the e-frame; and $\boldsymbol{\omega}_{ie}^{e}$ is the angular rate between the e-frame and the i-frame.

The algorithm in this paper aims at the low-cost MEMS-IMU, which has a gyro bias stability that is relatively large (10°/h in this paper) and is incapable of sensing the Earth's rotation (15°/h). Therefore, the Earth's rotation can be ignored, i.e., assuming $\boldsymbol{\omega}_{ie}^{e} = 0$, and Equation (4) is projected to the m-frame as follows:

$$\dot{\mathbf{v}}^{m} = \mathbf{C}_{b}^{m}\mathbf{f}^{b} + \mathbf{g}^{m} \tag{4}$$

After the integral of Equation (4), the velocity update equation is obtained:

$$\mathbf{v}_{b(t_k)}^{m} = \mathbf{v}_{b(t_{k-1})}^{m} + \int_{t_{k-1}}^{t_k} \left[C_{b}^{m}(t) f^{b}(t)\right] dt + \int_{t_{k-1}}^{t_k} g^{m} dt \tag{5}$$

where $\mathbf{v}_{b(t_k)}^{m}$ is the velocity of the IMU in the m-frame at t_k.

For vehicles travelling at medium or low speeds (less than 100 m/s), the angular rate and specific force can be assumed to be constant during the IMU sampling interval; that is, according to the single sampling hypothesis [25], the discrete form of Equation (5) is as follows:

$$\begin{aligned} \mathbf{v}_{b(t_k)}^{m} &= \mathbf{v}_{b(t_{k-1})}^{m} + \mathbf{C}_{b(t_{k-1})}^{m} \Delta\mathbf{v}_{b(t_k)}^{b(t_{k-1})} + \mathbf{g}^{m}\Delta t_k \\ \Delta\mathbf{v}_{b(t_k)}^{b(t_{k-1})} &= \Delta\mathbf{v}_{f,t_k}^{b} + \frac{1}{2}\Delta\boldsymbol{\theta}_{t_k} \times \Delta\mathbf{v}_{f,t_k}^{b} \end{aligned} \tag{6}$$

where $\Delta\mathbf{v}_{f,t_k}^{b}$ and $\Delta\boldsymbol{\theta}_{t_k}$ are the IMU outputs in incremental form with a bias compensation from t_{k-1} to t_k.

$$\begin{aligned} \Delta\mathbf{v}_{f,t_k}^{b} &= \int_{t_{k-1}}^{t_k} \left[\widetilde{f}^{b}(t) - b_a(t)\right] dt \\ \Delta\boldsymbol{\theta}_{t_k} &= \int_{t_{k-1}}^{t_k} \left[\widetilde{\omega}_{ib}^{b}(t) - b_g(t)\right] dt \end{aligned} \tag{7}$$

where $\mathbf{b}_a$ and $\mathbf{b}_g$ are the biases of the accelerometer and gyroscope, respectively.

2. Position Update

The position differential equation in the m-frame is as follows:

$$\dot{\mathbf{p}}^{m} = \mathbf{v}^{m} \tag{8}$$

The velocity at t_k has been updated, so the position update can be accomplished with the average velocity:

$$\mathbf{p}^m_{b(t_k)}=\mathbf{p}^m_{b(t_{k-1})}+\mathbf{v}^m_{b(t_{k-1})}\Delta t_k+\frac{1}{2}\mathbf{g}^m\Delta t_k^2+\frac{1}{2}\mathbf{C}^m_{b(t_{k-1})}\boldsymbol{\Delta}\mathbf{v}^{b(t_{k-1})}_{b(t_k)}\Delta t_k \tag{9}$$

where $\mathbf{p}^m_{b(t_k)}$ is the position of the IMU in the m-frame at t_k.

3. Attitude Update

The attitude is updated by the quaternion as follows [24]:

$$\begin{aligned}\mathbf{q}^m_{b(t_k)}&=\mathbf{q}^m_{b(t_{k-1})}\otimes\mathbf{q}^{b(t_{k-1})}_{b(t_k)}\\ \mathbf{q}^{b(t_{k-1})}_{b(t_k)}&=\begin{bmatrix}\cos\|0.5\boldsymbol{\Phi}_{t_k}\|\\ \frac{\sin\|0.5\boldsymbol{\Phi}_{t_k}\|}{\|\boldsymbol{\Phi}_{t_k}\|}\boldsymbol{\Phi}_{t_k}\end{bmatrix}\end{aligned} \tag{10}$$

where $\mathbf{q}^m_{b(t_k)}$ is the quaternion from the b-frame to the m-frame at t_k; and $\boldsymbol{\Phi}_{t_k}$ is the equivalent rotation vector of the b-frame from t_{k-1} to t_k. According to the single sampling hypothesis [25], we have the following equation:

$$\boldsymbol{\Phi}_{t_k}\approx\Delta\boldsymbol{\theta}_{t_k} \tag{11}$$

2.2.2. Local Registration

The LiDAR (VLP-16) used in this paper is a mechanical rotary system with a sampling rate of 300–1200 RPM (revolutions per minute). The interval of the two adjacent data packages is approximately 1.3 ms, and one circle of the point cloud contains approximately 154–38 packages. A point cloud circle is used in the scan matching. Therefore, the point cloud data packages need to be spliced to form a circle of point clouds based on the sampling time of a certain package.

Because the amount of point cloud data is relatively large, which affects the speed of point cloud splicing and scan matching, the voxel filter [26] is needed for point cloud downsampling.

Considering the vehicle motion during point cloud collection, motion compensation is required during splicing: here linear interpolation is used, given the factor $f=\frac{t_n-t_{k-1}}{t_k-t_{k-1}}(t_{k-1}\le t_n\le t_k)$, $\omega=\arccos\left(\mathbf{q}^m_{b(t_{k-1})}\cdot\mathbf{q}^m_{b(t_k)}\right)$, then the position and attitude in the m-frame of the data package at t_n is interpolated according to Equation (12) [27] with the position and attitude of the IMU in the m-frame that was obtained in Section 2.2.1. Then, a start time for splicing is selected, and the position and attitude of the data packages within one rotation range are transformed into the relative position and attitude to the start time, as well as the corresponding point cloud. Finally, one circle of the point cloud with the motion compensation, which is recorded as a node, is obtained. The flow of the local registration is shown in Figure 2.

$$\begin{aligned}\mathbf{R}^m_{b(t_n)}&=\left(\mathbf{p}^m_{b(t_{k-1})}+\left(\mathbf{p}^m_{b(t_k)}-\mathbf{p}^m_{b(t_{k-1})}\right)f,\ \frac{\sin(1-f)\omega}{\sin\omega}\mathbf{q}^m_{b(t_{k-1})}+\frac{\sin f\omega}{\sin\omega}\mathbf{q}^m_{b(t_k)}\right)\\ \mathbf{R}^m_{l(t_n)}&=\mathbf{R}^m_{b(t_n)}\mathbf{R}^b_l\end{aligned} \tag{12}$$

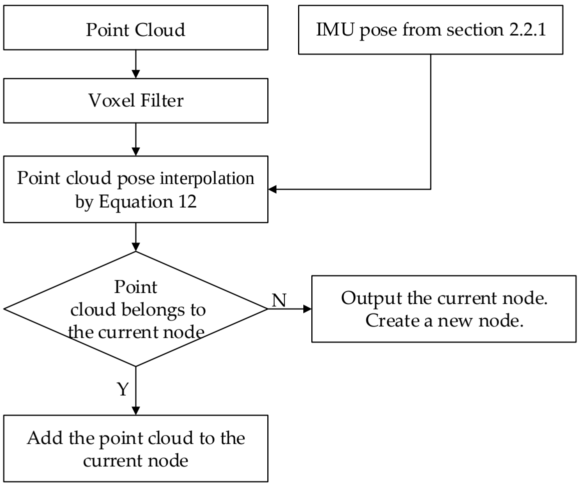
Point Cloud
IMU pose from section 2.2.1
Voxel Filter
Point cloud pose interpolation by Equation 12
Point cloud belongs to the current node
N
Output the current node. Create a new node.
Y
Add the point cloud to the current node

Figure 2. Flow chart of local registration.

2.2.3. Probability Map

When the scan point is added to the probability map, the scan point is converted into a grid point (such as the dark grey square in Figure 3), which has a probability value that is set as p_{hit}. The points on the line that connect the scan point and the scan origin, which has a probability value that is set as p_{miss}, are also converted into grid points (such as the light grey squares in Figure 3). To reduce the calculation load, only parts of the points on the line are converted. If a grid point has been assigned, the probability value is updated as Equation (13) [28]:

$$\begin{aligned} odds(p) &= \tfrac{p}{1-p} \\ Map_{new}(\mathbf{x}) &= odds^{-1}[odds(Map_{old}(\mathbf{x})) \times odds(p)] \end{aligned} \quad (13)$$

where *Map* is the mapping function of the coordinates for the probability value of the point in the probability map; and **x** represents the coordinates of the point cloud in the m-frame.

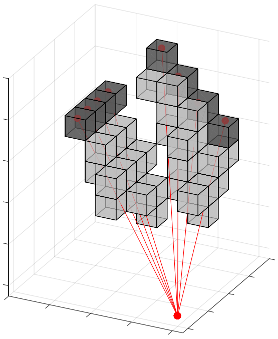

Figure 3. Schematic diagram of the probability map.

For the convenience of expressing a wide ranging map, as well as the later optimization of the pose, multiple submaps, each of which includes a probability map and the pose of the probability map in the m-frame, are applied in this paper.

In the front-end scan matching period, two active submaps (such as the $k-1$th and kth submaps in Figure 4) are maintained. The nodes (such as the green nodes in Figure 4) are matched with the $k-1$th submap, and then, those nodes will be simultaneously inserted into the $k-1$th and kth submaps. When the number of nodes added in the kth submap reaches a certain threshold n, the $k-1$th submap whose probability value will not be updated is fixed. The subsequent nodes (blue nodes in Figure 4) will be added to the kth submap, while the $k+1$th submap is created with the pose of the n+1th node in the kth submap. The blue nodes are also added to the $k+1$th submap. Thus, there are always two active submaps, and half of the areas in the two adjacent submaps overlap.

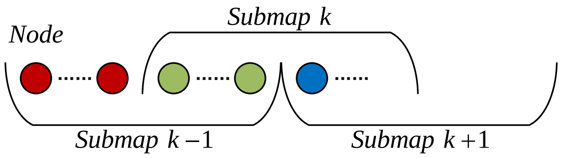

Figure 4. Schematic diagram of submaps and nodes.

The value of the threshold n directly affects the precision in the front-end scan matching and the number of parameters in the back-end graph optimization. On the one hand, if it is too small, the submap range will be too small, the probability map will be inaccurate in expression, and therefore, the scan matching accuracy will be reduced. The number of submap parameters in the back-end graph optimization will increase, which affects the optimization efficiency. On the other hand, if the threshold is too large, the submap range will be too large, the probability map will be incorrect in the expression because submap formation is mainly through the front-end scan matching of the probability map, and the matching accuracy will decrease as the mileage distance increases. Therefore, the threshold should be moderate. In addition, the value of this threshold is directly related to the vehicle's speed and sampling rate of LiDAR. In this paper, the vehicle's speed is approximately 10 m/s, and the sampling rate of LiDAR is 600 RPM, while the parameter n is set to 100; that is, a submap is created approximately every 10 m.

2.2.4. Nonlinear Correlative Scan Matching

The optimization pose can be obtained by solving the nonlinear least squares objective function (Equation (14)) with the initial pose of the node in the m-frame obtained in Section 2.2.2. Ceres [29] is used as the nonlinear optimization solver in this paper.

$$\underset{\mathbf{R}_l^m}{\operatorname{argmin}} \sum_{k=1}^{K} \left(1 - Map_{smooth}\left(R_l^m p_k^l\right)\right)^2 \tag{14}$$

where K is the point number contained in the node; and $\mathbf{p}_k^l$ is the coordinate of the kth point in the l-frame; and Map_{smooth} is a smooth version of the Map by the tricubic interpolation [30].

To reduce the number of nodes inserted into the probability map when the vehicle moves slowly or in static, the pose of the current node is compared with the pose of the previous node inserted into the probability map. Only when a certain pose or time change threshold is achieved will the node be added to the probability map. In this paper, the thresholds of the changes in displacement, yaw and time are 0.5 m, 1° and 0.5 s, respectively.

2.3. Relative Pose Estimation

The estimation of relative pose, solving the relative pose between all nodes and relevant submaps (the node can be successfully matched in the submap), offers constraints for graph optimization. To reduce the number of nodes, the nodes participating in graph optimization are selected from those inserted into the probability map in Section 2.2.4 with a time interval of 1 s. When a node participates in the construction of the searched submap, the relative pose between the node and submap is calculated directly by the node pose and the submap pose in the m-frame obtained in the front-end. Otherwise, scan matching is performed again in the submap. If successfully matched, then regard the node as a closed-loop and increase its weight in the graph optimization. The loss function is added to avoid a false loop. The distance search step is the resolution of the submap, and the yaw search step is the angle at which one grid is rotated by the longest distance of the points in the node. Within the distance and yaw search thresholds, a set of possible position searches and yaw searches is formed according to the search step. Because the number of the set is relatively large, to quickly find the most suitable solution, the branch and bound method [15,31] is adopted. Similar to Section 2.2.4, the result of the search matching is taken as the initial value, the final node pose is obtained, and the relative displacement and attitude of the node in the submap are further obtained.

2.4. Graph Optimization

The nonlinear optimization method is adopted in the back-end, with the parameter to be estimated being χ (Equation (15)).

$$\begin{aligned} \chi &= \left[\mathbf{x}_{t_1}, \mathbf{x}_{t_2} \cdots \mathbf{x}_{t_N}, \mathbf{y}_{t_1}, \mathbf{y}_{t_2} \cdots \mathbf{y}_{t_M}, \mathbf{R}_m^w\right] \\ \mathbf{x}_{t_k} &= \left[\mathbf{R}_{b(t_k)}^m, \mathbf{v}_{b(t_k)}^m, \mathbf{b}_a(t_k), \mathbf{b}_g(t_k)\right], k \in [1, N] \\ \mathbf{y}_{t_k} &= \left[\mathbf{R}_{s(t_k)}^m\right], k \in [1, M] \end{aligned} \tag{15}$$

where $\mathbf{x}_{t_k}$ is composed of the pose and velocity of the IMU in the m-frame and the bias of the accelerometer and gyroscope at t_k; N is the number of nodes; $\mathbf{y}_{t_k}$ is the pose of submap in the m-frame at t_k; and M is the number of submaps.

The cost functions are as follows:

$$\underset{\chi}{\operatorname{argmin}} \left\{ \begin{array}{c} \sum\limits_{i \in \alpha} E_{gnss}^2\left(x_{t_i}, R_m^w, p_{g(t_i)}^w, l_g^b, \sigma_p\right) + \sum\limits_{k=1}^{N-1} E_{imu}^2\left(x_{t_k}, x_{t_{k+1}}, z_{t_{k+1}}^{t_k}, \sigma_z\right) \\ + \sum\limits_{\substack{i \in \beta \\ j \in \kappa}} E_{lidar}^2\left(x_{t_i}, y_{t_j}, R_l^b, R_{l(t_i)}^{s(t_j)}, \sigma_{ij}\right) \end{array} \right\} \tag{16}$$

where E_{gnss}^2 is the GNSS cost function; $\mathbf{p}_g^w$ is the GNSS positioning result in the w-frame; $\mathbf{l}_g^b$ is the lever arm of the GNSS antenna; $\boldsymbol{\sigma}_p$ is the standard deviation of $\mathbf{p}_g^w$; α is the set of nodes with the GNSS positioning result; E_{imu}^2 is the IMU preintegration cost function; $\mathbf{z}_{t_{k+1}}^{t_k}$ is the preintegration result between the t_k node and the t_{k+1} node; $\boldsymbol{\sigma}_z$ is the standard deviation of $\mathbf{z}$; E_{lidar}^2 is the LiDAR cost function; $\mathbf{R}_{l(t_i)}^{s(t_j)}$ is the relative pose of the node $l(t_i)$ and the submap $s\left(t_j\right)$; $\boldsymbol{\sigma}_{ij}$ is the standard deviation of $\mathbf{R}_{l(t_i)}^{s(t_j)}$; β is the set of nodes; and κ is the set of submaps.

2.4.1. Sensor Calibration

The sensor calibration is important in multi-sensor fusion. The lever arm of the GNSS antenna $\mathbf{l}_g^b$ and the translation $\mathbf{p}_l^b$ between the IMU and the LiDAR are measured with a tape measure. For the quaternion $\mathbf{q}_l^b$ between the IMU and the LiDAR, it is added to χ and estimated according to Equation (32) by using approximately 15 min of data in good GNSS signal area.

2.4.2. GNSS Cost Function

The latitude, longitude and height given by the GNSS solution are first converted into the e-frame coordinates. Then, the e-frame coordinates are converted into those in the w-frame [32]. Finally, the absolute position modified cost function of the GNSS at t_i is formed in the w-frame as Equation (17):

$$\begin{aligned} E_{gnss}^2\left(\mathbf{x}_{t_i}, \mathbf{R}_m^w, \mathbf{p}_{g(t_i)}^w, \mathbf{l}_g^b, \boldsymbol{\sigma}_p\right) &= e\left(\mathbf{x}_{t_i}, \mathbf{R}_m^w, \mathbf{p}_{g(t_i)}^w, \mathbf{l}_g^b\right)^T \left(\boldsymbol{\sigma}_p^2\right)^{-1} e\left(\mathbf{x}_{t_i}, \mathbf{R}_m^w, \mathbf{p}_{g(t_i)}^w, \mathbf{l}_g^b\right) \\ e\left(\mathbf{x}_{t_i}, \mathbf{R}_m^w, \mathbf{p}_{g(t_i)}^w, \mathbf{l}_g^b\right) &= \mathbf{R}_m^w\left(\mathbf{R}_{b(t_i)}^m \mathbf{l}_g^b\right) - \mathbf{p}_{g(t_i)}^w \end{aligned} \tag{17}$$

In this paper, the sampling rate of the GNSS is 1 Hz, so the GNSS solution result is at the integer second moment. To make **x** exist at the integer second moment, the data package close to the integer second moment is selected as the splicing starting point for the point cloud splicing in Section 2.2.2, and the difference between the package time and the integer second moment is ignored.

2.4.3. IMU Preintegration Cost Function

According to Equations (6), (9) and (10), the recursion relationship of the position, velocity and attitude of the IMU in the m-frame between t_{i-1} and t_i can be obtained as Equation (18):

$$\begin{aligned}\mathbf{v}^m_{b(t_i)} &= \mathbf{v}^m_{b(t_{i-1})} + \mathbf{C}^m_{b(t_{i-1})}\mathbf{v}^{b(t_{i-1})}_{b(t_i)} + \mathbf{g}^m\Delta t_i \\ \mathbf{p}^m_{b(t_i)} &= \mathbf{p}^m_{b(t_{i-1})} + \mathbf{v}^m_{b(t_{i-1})}\Delta t_i + \tfrac{1}{2}\mathbf{g}^m\Delta t_i^2 + \tfrac{1}{2}\mathbf{C}^m_{b(t_{i-1})}\mathbf{v}^{b(t_{i-1})}_{b(t_i)}\Delta t_i \\ \mathbf{q}^m_{b(t_i)} &= \mathbf{q}^m_{b(t_{i-1})} \otimes \mathbf{q}^{b(t_{i-1})}_{b(t_i)}\end{aligned} \tag{18}$$

Then, the preintegration form (Equation (19)) is built up according to Equation (18) [9].

$$\begin{aligned}\mathbf{v}^{b(t_{i-1})}_{b(t_i)} &= \mathbf{C}^{b(t_{i-1})}_m\mathbf{v}^m_{b(t_i)} - \mathbf{C}^{b(t_{i-1})}_m\left(\mathbf{v}^m_{b(t_{i-1})} + \mathbf{g}^m\Delta t_i\right) \\ \mathbf{p}^{b(t_{i-1})}_{b(t_i)} &= \mathbf{C}^{b(t_{i-1})}_m\mathbf{p}^m_{b(t_i)} - \mathbf{C}^{b(t_{i-1})}_m\left(\mathbf{p}^m_{b(t_{i-1})} + \mathbf{v}^m_{b(t_{i-1})}\Delta t_i + \tfrac{1}{2}\mathbf{g}^m\Delta t_i^2\right) \\ \mathbf{q}^{b(t_{i-1})}_{b(t_i)} &= \mathbf{q}^{b(t_{i-1})}_m \otimes \mathbf{q}^m_{b(t_i)}\end{aligned} \tag{19}$$

where $\mathbf{p}^{b(t_{i-1})}_{b(t_i)} = \frac{1}{2}\mathbf{v}^{b(t_{i-1})}_{b(t_i)}\Delta t_i$, $\mathbf{v}^{b(t_{i-1})}_{b(t_i)}$ and $\mathbf{q}^{b(t_{i-1})}_{b(t_i)}$ are the preintegration form of the increments in position, velocity, and attitude, which are independent of the pose at the starting point of the integration and are only related to the original output and the IMU bias. Therefore, the preintegration form of the increments in position, velocity, and attitude between the t_{k-1} node and the t_k node can be obtained by the recursive formula Equation (20):

$$\begin{aligned}\mathbf{v}^{b(t_{k-1})}_{b(t_{k-1})} &= 0, \mathbf{p}^{b(t_{k-1})}_{b(t_{k-1})} = 0, \mathbf{q}^{b(t_{k-1})}_{b(t_{k-1})} = \mathbf{I} \\ \mathbf{v}^{b(t_{k-1})}_{b(t_{i+1})} &= \mathbf{v}^{b(t_{k-1})}_{b(t_i)} + \mathbf{C}^{b(t_{k-1})}_{b(t_i)}\mathbf{v}^{b(t_i)}_{b(t_{i+1})} \\ \mathbf{p}^{b(t_{k-1})}_{b(t_{i+1})} &= \mathbf{p}^{b(t_{k-1})}_{b(t_i)} + \tfrac{1}{2}\mathbf{C}^{b(t_{k-1})}_{b(t_i)}\mathbf{v}^{b(t_i)}_{b(t_{i+1})}\Delta t_{i+1} \\ \mathbf{q}^{b(t_{k-1})}_{b(t_{i+1})} &= \mathbf{q}^{b(t_{k-1})}_{b(t_i)} \otimes \mathbf{q}^{b(t_i)}_{b(t_{i+1})}\end{aligned} \tag{20}$$

where t_i is the IMU sampling moment between t_{k-1} and t_k, $t_i \in [t_{k-1}, t_k]$.

The IMU data at t_{k-1} and t_k are obtained by linear interpolation. Assuming that the bias between the two adjacent nodes does not change, the original outputs of the IMU between t_{k-1} and t_k are compensated for using the bias at t_{k-1} by Equation (7).

So the preintegration result $\mathbf{z}^{t_{k-1}}_{t_i}$ is showed as Equation (21):

$$\mathbf{z}^{t_{k-1}}_{t_i} = \begin{bmatrix} \mathbf{p}^{b(t_{k-1})}_{b(t_i)} & \mathbf{v}^{b(t_{k-1})}_{b(t_i)} & \mathbf{q}^{b(t_{k-1})}_{b(t_i)} \end{bmatrix}^T \tag{21}$$

Then the covariance of $\mathbf{z}^{t_{k-1}}_{t_k}$ is discussed. Because $\mathbf{q}^{b(t_{k-1})}_{b(t_i)}$ is over-parameterized, its error term can be defined as Equation (22) [9]:

$$\mathbf{q}^{b(t_{k-1})}_{b(t_i)} \approx \hat{\mathbf{q}}^{b(t_{k-1})}_{b(t_i)} \otimes \begin{bmatrix} 1 \\ \frac{1}{2}\delta\boldsymbol{\theta}^{b(t_{k-1})}_{b(t_i)} \end{bmatrix} \tag{22}$$

where $\delta\boldsymbol{\theta}^{b(t_{k-1})}_{b(t_i)}$ is the error of $\hat{\mathbf{q}}^{b(t_{k-1})}_{b(t_i)}$ in the form of equivalent rotation vector.

The accelerometer observation error $\delta\mathbf{f}^b$ and gyroscope observation error $\delta\boldsymbol{\omega}_{ib}^b$ include the bias and white noise:

$$\begin{aligned}\delta\mathbf{f}^b &= \mathbf{b}_a + \mathbf{w}_a \\ \delta\boldsymbol{\omega}_{ib}^b &= \mathbf{b}_g + \mathbf{w}_g\end{aligned} \tag{23}$$

where $\mathbf{w}_a$ and $\mathbf{w}_g$ are the white noises of the accelerometer and gyroscope observations, respectively.

The bias error is modeled as the first-order Gauss Markov process [25], which is shown as Equation (24):

$$\begin{aligned}\delta\dot{\mathbf{b}}_a &= -\delta\mathbf{b}_a/\tau_a + \mathbf{w}_{ca} \\ \delta\dot{\mathbf{b}}_g &= -\delta\mathbf{b}_g/\tau_g + \mathbf{w}_{cg}\end{aligned} \tag{24}$$

where τ and $\mathbf{w}_c$ are the correlation time and white noise of the first-order Gauss Markov process, respectively.

The differential equation for the error terms of the preintegration result can be derived from Equation (20) and Equation (24) [33], which is shown as Equation (26):

$$\delta\dot{\mathbf{z}}_{t_i}^{t_{k-1}} = \mathbf{F}(t_i)\delta\mathbf{z}_{t_i}^{t_{k-1}} + \mathbf{G}(t_i)\mathbf{w}(t_i) \tag{25}$$

where

$$\begin{aligned}
\mathbf{F}(t_i) &= \begin{bmatrix} 0 & \mathbf{I} & 0 & 0 & 0 \\ 0 & 0 & -\mathbf{C}_{b(t_i)}^{b(t_{k-1})}\left[\mathbf{f}^{b(t_i)}\times\right] & \mathbf{C}_{b(t_i)}^{b(t_{k-1})} & 0 \\ 0 & 0 & -\left[\boldsymbol{\omega}_{ib(t_i)}^{b(t_i)}\times\right] & 0 & \mathbf{I} \\ 0 & 0 & 0 & -1/\tau_a & 0 \\ 0 & 0 & 0 & 0 & -1/\tau_g \end{bmatrix} \\
\delta\mathbf{z}_{t_i}^{t_{k-1}} &= \begin{bmatrix} \delta\mathbf{p}_{b(t_i)}^{b(t_{k-1})} & \delta\mathbf{v}_{b(t_i)}^{b(t_{k-1})} & \delta\boldsymbol{\theta}_{b(t_i)}^{b(t_{k-1})} & \delta\mathbf{b}_a & \delta\mathbf{b}_g \end{bmatrix}^T \\
\mathbf{G}(t_i) &= \begin{bmatrix} 0 & 0 & 0 & 0 \\ \mathbf{C}_{b(t_i)}^{b(t_{k-1})} & 0 & 0 & 0 \\ 0 & \mathbf{I} & 0 & 0 \\ 0 & 0 & \mathbf{I} & 0 \\ 0 & 0 & 0 & \mathbf{I} \end{bmatrix} \\
\mathbf{w}(t_i) &= \begin{bmatrix} \mathbf{w}_a & \mathbf{w}_g & \mathbf{w}_{ca} & \mathbf{w}_{cg} \end{bmatrix}^T
\end{aligned} \tag{26}$$

The discrete form of Equation (25) is Equation (27) [34]:

$$\delta\mathbf{z}_{t_{i+1}}^{t_{k-1}} = \boldsymbol{\Phi}_{i+1,i}\delta\mathbf{z}_{t_i}^{t_{k-1}} + \mathbf{W}_{t_i} \tag{27}$$

where

$$\begin{aligned}\boldsymbol{\Phi}_{i+1,i} &\approx \mathbf{F}(t_{i+1})\Delta t_{i+1} + \mathbf{I} \\ \mathbf{W}_{t_i} &= \int_{t_i}^{t_{i+1}} \Phi_{i+1,\tau}G(\tau)w(\tau)d\tau\end{aligned} \tag{28}$$

The variance matrix $\mathbf{P}_{t_{k-1}}$ of $\delta\mathbf{z}_{t_{k-1}}^{t_{k-1}}$ is set to 0, and $\mathbf{P}$ is updated as follows [34]:

$$\begin{aligned}\mathbf{P}_{t_{i+1}} &= \boldsymbol{\Phi}_{i,i-1}\mathbf{P}_{t_i}\boldsymbol{\Phi}_{i,i-1}^T + \mathbf{Q}_{t_i} \\ \mathbf{Q}_{t_i} &= \frac{1}{2}\left[\boldsymbol{\Phi}_{i,i-1}\mathbf{G}(t_{i-1})\mathbf{q}_{t_{i-1}}\mathbf{G}(t_{i-1})^T + \mathbf{G}(t_{i-1})\mathbf{q}_{t_{i-1}}\mathbf{G}(t_{i-1})^T\boldsymbol{\Phi}_{i,i-1}^T\right]\Delta t_i\end{aligned} \tag{29}$$

where $\mathbf{Q}$ is the variance matrix of $\mathbf{W}$; and $\mathbf{q}$ is the variance matrix of $\mathbf{w}$.

When the bias estimate changes slightly, a first-order approximation of $\mathbf{p}_{b(t_k)}^{b(t_{k-1})}$, $\mathbf{v}_{b(t_k)}^{b(t_{k-1})}$ and $\mathbf{q}_{b(t_k)}^{b(t_{k-1})}$ can be used to correct the preintegration result as Equation (30) instead of re-propagation [9].

$$\begin{aligned} \tilde{\mathbf{v}}_{b(t_k)}^{b(t_{k-1})} &= \mathbf{v}_{b(t_k)}^{b(t_{k-1})} + \mathbf{\Phi}_{\mathbf{b}_a}^{\mathbf{v}} \delta \mathbf{b}_a + \mathbf{\Phi}_{\mathbf{b}_g}^{\mathbf{v}} \delta \mathbf{b}_g \\ \tilde{\mathbf{p}}_{b(t_k)}^{b(t_{k-1})} &= \mathbf{p}_{b(t_k)}^{b(t_{k-1})} + \mathbf{\Phi}_{\mathbf{b}_a}^{\mathbf{p}} \delta \mathbf{b}_a + \mathbf{\Phi}_{\mathbf{b}_g}^{\mathbf{p}} \delta \mathbf{b}_g \\ \tilde{\mathbf{q}}_{b(t_k)}^{b(t_{k-1})} &= \mathbf{q}_{b(t_k)}^{b(t_{k-1})} \otimes \begin{bmatrix} 1 \\ \frac{1}{2} \mathbf{\Phi}_{\mathbf{b}_g}^{\mathbf{q}} \delta \mathbf{b}_g \end{bmatrix} \end{aligned} \tag{30}$$

where $\mathbf{\Phi}_{\mathbf{b}_a}^{\mathbf{v}}$ is the sub-block matrix in $\mathbf{\Phi}_{k,k-1}$ whose location is corresponding to $\frac{\delta \mathbf{v}}{\delta \mathbf{b_a}}$.The same meaning is also used for $\mathbf{\Phi}_{\mathbf{b}_g}^{\mathbf{v}}$, $\mathbf{\Phi}_{\mathbf{b}_a}^{\mathbf{p}}$, $\mathbf{\Phi}_{\mathbf{b}_g}^{\mathbf{p}}$, $\mathbf{\Phi}_{\mathbf{b}_g}^{\mathbf{q}}$.

Finally, the IMU preintegration cost function is obtained with Equation (31):

$$\begin{gathered} E_{imu}^2\left(\mathbf{x}_{t_{k-1}}, \mathbf{x}_{t_k}, \mathbf{z}_{t_k}^{t_{k-1}}, \mathbf{\sigma}_z\right) = e\left(\mathbf{x}_{t_{k-1}}, \mathbf{x}_{t_k}, \mathbf{z}_{t_k}^{t_{k-1}}\right)^T \left(\mathbf{\sigma}_z^2\right)^{-1} e\left(\mathbf{x}_{t_{k-1}}, \mathbf{x}_{t_k}, \mathbf{z}_{t_k}^{t_{k-1}}\right) \\ e\left(\mathbf{x}_{t_{k-1}}, \mathbf{x}_{t_k}, \mathbf{z}_{t_k}^{t_{k-1}}\right) = \begin{bmatrix} \mathbf{C}_m^{b(t_{k-1})}\left(\mathbf{p}_{b(t_k)}^m - \mathbf{p}_{b(t_{k-1})}^m - \mathbf{v}_{b(t_{k-1})}^m \Delta t_k + \frac{1}{2}\mathbf{g}^m \Delta t_k^2\right) - \tilde{\mathbf{p}}_{b(t_k)}^{b(t_{k-1})} \\ \mathbf{C}_m^{b(t_{k-1})}\left(\mathbf{v}_{b(t_k)}^m - \mathbf{v}_{b(t_{k-1})}^m + \mathbf{g}^m \Delta t_k\right) - \tilde{\mathbf{v}}_{b(t_k)}^{b(t_{k-1})} \\ 2\left[\left(\left(\mathbf{q}_{b(t_k)}^m\right)^{-1} \otimes \mathbf{q}_{b(t_{k-1})}^m\right) \otimes \tilde{\mathbf{q}}_{b(t_k)}^{b(t_{k-1})}\right]_{xyz} \\ \mathbf{b}_a(t_k) - \mathbf{b}_a(t_{k-1}) \\ \mathbf{b}_g(t_k) - \mathbf{b}_g(t_{k-1}) \end{bmatrix} \end{gathered} \tag{31}$$

where $\mathbf{\sigma}_z^2$ is the variance covariance matrix of the preintegration variable, $\mathbf{\sigma}_k^2 = \mathbf{P}_{t_k}$. $[\mathbf{q}]_{\mathrm{xyz}}$ is the equivalent rotation vector of $\mathbf{q}$.

2.4.4. LiDAR Cost Function

The relative pose between the node $l(t_i)$ and the submap $s\left(t_j\right)$ and variance matrix are calculated in Section 2.3. The LiDAR cost function is obtained with Equation (32) [22]:

$$\begin{gathered} E_{lidar}^2\left(\mathbf{x}_i, \mathbf{y}_j, \mathbf{R}_l^b, \mathbf{R}_{l(t_i)}^{s(t_j)}, \mathbf{\sigma}_{ij}\right) = e\left(\mathbf{x}_i, \mathbf{y}_j, \mathbf{R}_l^b, \mathbf{R}_{l(t_i)}^{s(t_j)}\right)^T \left(\mathbf{\sigma}_{ij}^2\right)^{-1} e\left(\mathbf{x}_i, \mathbf{y}_j, \mathbf{R}_l^b, \mathbf{R}_{l(t_i)}^{s(t_j)}\right) \\ e\left(\mathbf{x}_i, \mathbf{y}_j, \mathbf{R}_l^b, \mathbf{R}_{l(t_i)}^{s(t_j)}\right) = \begin{bmatrix} \left(\mathbf{C}_{s(t_j)}^m\right)^{-1}\left[\mathbf{p}_{b(t_i)}^m + \mathbf{R}_{b(t_i)}^m \mathbf{p}_l^b - \mathbf{p}_{s(t_j)}^m\right] - \mathbf{p}_{l(t_i)}^{s(t_j)} \\ \left[\left(\mathbf{q}_{b(t_i)}^m \otimes \mathbf{q}_l^b\right)^{-1} \otimes \mathbf{q}_{s(t_j)}^m \otimes \mathbf{q}_{l(t_i)}^{s(t_j)}\right]_{xyz} \end{bmatrix} \end{gathered} \tag{32}$$

2.4.5. Sliding Window

In the back-end graph optimization, the submaps and nodes will remarkably increase with time, considering the amount of optimization calculation. Therefore, it is necessary to limit the number of optimization variables and selectively remove some historical variables while adding new variables.

The sliding window [35] is applied in this paper to ensure that the computation amount does not increase as the optimization variable increases. A fixed number of submaps and variables of related nodes are saved in the sliding window. When a submap is added, the oldest submap and related node variables in the sliding window are discarded; thus, the number of submaps in the sliding window remains fixed.

However, if the variables are directly removed, there will be a loss of information, and the nodes associated with the discarded submap may also be related to other submaps in the sliding window. Therefore, the direct discarding method eliminates these constraints.

The cost function **e** in the graph optimization is nonlinear, and the nonlinear least squares problem can be obtained by linear iteration as follows:

$$\mathbf{H}\delta\chi = \mathbf{y} \tag{33}$$

where $\mathbf{H}=\mathbf{J}^{\mathbf{T}}\mathbf{W}\mathbf{J}$ and $\mathbf{y}=\mathbf{J}\mathbf{W}\mathbf{e}$; **J** is the Jacobian matrix; and **W** is the weight matrix.

$$\begin{bmatrix} \mathbf{H}_a & \mathbf{H}_b \\ \mathbf{H}_b^T & \mathbf{H}_c \end{bmatrix}\begin{bmatrix} \delta\chi_a \\ \delta\chi_b \end{bmatrix} = \begin{bmatrix} \mathbf{y}_a \\ \mathbf{y}_b \end{bmatrix} \tag{34}$$

χ_a must be marginalized, while χ_b is retained. After Scherer's elimination [36], we obtain Equation (35).

$$\begin{gathered} \mathbf{H}_d = \mathbf{H}_c - \mathbf{H}_b^T\mathbf{H}_a^{-1}\mathbf{H}_b \\ \begin{bmatrix} \mathbf{H}_a & \mathbf{H}_b \\ 0 & \mathbf{H}_d \end{bmatrix}\begin{bmatrix} \delta\chi_a \\ \delta\chi_b \end{bmatrix} = \begin{bmatrix} \mathbf{y}_a \\ \mathbf{y}_b - \mathbf{H}_b^T\mathbf{H}_a^{-1}\mathbf{y}_a \end{bmatrix} \end{gathered} \tag{35}$$

According to Equation (35), the residual $\delta\chi_b$ of the reserved variable χ_b can be obtained. In this process, the information of the marginalized variable χ_a is utilized; that is, the constraint is not discarded, and a marginalize cost function (Equation (36)) is added to the graph optimization to introduce constraints on marginalized variables [37].

$$\begin{gathered} E_{marg}^2\left(\mathbf{H}_d, \chi_b, \tilde{\chi}_b, \delta\chi_b\right) = e\left(\mathbf{H}_d, \chi_b, \tilde{\chi}_b, \delta\chi_b\right)^T e\left(\mathbf{H}_d, \chi_b, \tilde{\chi}_b, \delta\chi_b\right) \\ e\left(\mathbf{H}_d, \chi_b, \tilde{\chi}_b, \delta\chi_b\right) = \mathbf{H}_d\left(\chi_b - \tilde{\chi}_b\right) + \delta\chi_b \end{gathered} \tag{36}$$

where $\tilde{\chi}_b$ is the estimated value of χ_b in the residual calculation.

The size of the sliding window affects the efficiency and precision of the graph optimization, and this sizing is necessary to ensure an adequate amount of submaps. The sliding window size used in this paper is 5.

3. Experiment

To verify the performance of the proposed GNSS/INS/LiDAR-SLAM integrated navigation system, land vehicle tests were conducted in Wuhan on September 7, 2018. In addition to VLP16 and the tested system (NV-POS1100), the vehicle was equipped with a higher precision inertial integrated navigation system (LD-A15) as the reference system in the tests, as shown in Figure 5. Table 1 gives the technical parameters of the two systems. The sampling rate of GNSS was 1 Hz, the sampling rate of LiDAR was 600 RPM, and the sampling rate of IMU was 200 Hz. All the test data has been shared on OneDrive (https://whueducn-my.sharepoint.com/:f:/g/personal/changlesgg_whu_edu_cn/EsN45ma2spBMmC37pafR3Q0BdeMuD_hb1uBc3gsQERu-uw?e=KOwhwe).

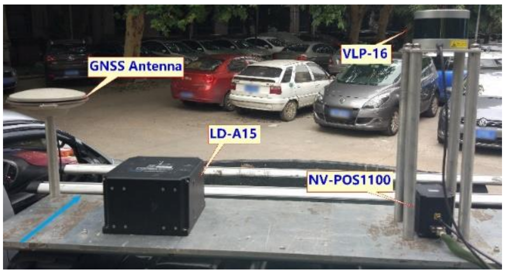

Figure 5. Experimental platform.

Table 1. Technical Parameters of LD-A15 and NV-POS1100.

IMU	Bias		Random Walk	
	Gyro. [°/h]	Acc. [mGal]	Angular [°/ $\sqrt{h}$]	Velocity [m/s/ $\sqrt{h}$]
LD-A15 (Reference system)	0.02	15	0.003	0.03
NV-POS1100 (Tested system)	10	1000	0.2	0.18

In the case of continuous GNSS position correction, the GNSS/INS integrated navigation system carries out observational updates with the GNSS position. The navigation accuracy, especially the position accuracy, is mainly determined by the GNSS positioning accuracy. Therefore, the accuracy assessment of the GNSS/INS integrated navigation system is achieved by investigating the navigation error during GNSS signal interruption. In this paper, the accuracy of the GNSS, IMU, and LiDAR-SLAM integrated navigation system was evaluated in the same way.

Three tests (approximately one hour for each test) were carried out in an open-sky environment, and one test was conducted in urban areas. The test trajectories are shown in Figures 6 and 7. The reference system (LD-A15) data were processed with the PPK (Post Processed Kinematic)/INS smoothing integration method. The results were converted to the center of the tested system (NV-POS1100) as the reference value of its position and attitude. Then, one minute of GNSS outage every six minutes was intentionally introduced into the PPK results of the reference system to simulate the GNSS signal interruption (all satellites' signals were lost and recovered at the same time). Thereafter, the GNSS/INS/LiDAR-SLAM integrated method described above was performed with the PPK result involving GNSS outages, the tested system data and the VLP16 data. For the purpose of comparison, the GNSS/INS integration method and the Cartographer's GNSS/LiDAR/IMU integrated navigation method were also conducted. Then, the navigation errors of the three methods during the GNSS outage were compared. Finally, the same comparative analysis was carried out with the data collected in urban areas, where signal attenuation and outage occur frequently, as shown in Figure 8. The performances of the three methods can be compared by checking the positioning drifts in the real GNSS signal outages and attenuations.

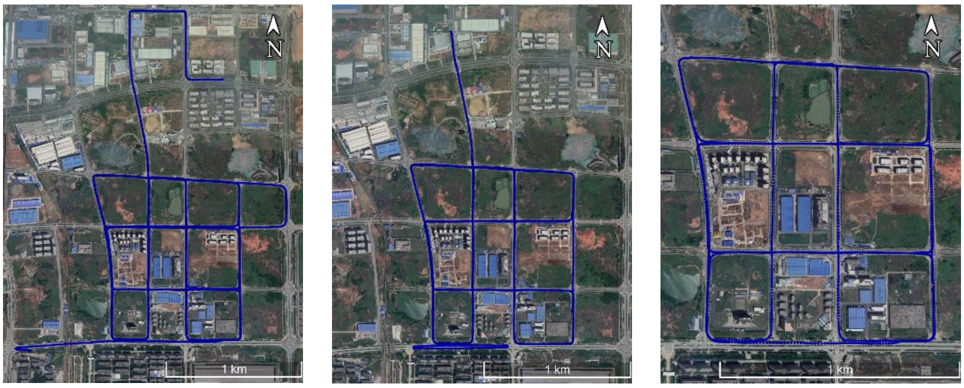

Figure a Test #1 Figure b Test #2 Figure c Test #3

Figure 6. Test trajectory in the open-sky environment.

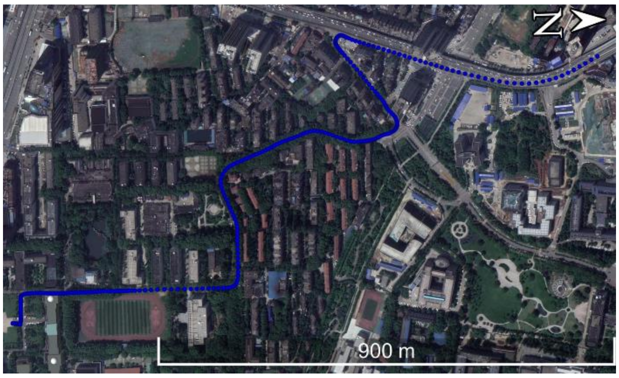

Figure 7. Test trajectory in the urban area.

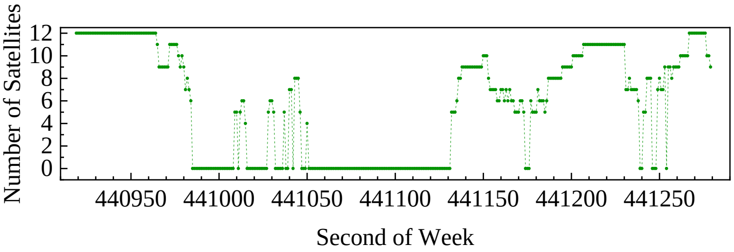

Figure 8. Number of satellites in the urban area.

4. Results and Discussion

The position and attitude errors of the three methods with the one minute's GNSS outage simulation test #2 are shown in Figures 9–11. The grey span in the figures marks the periods of simulating the GNSS outages. Based on the position and attitude errors in the GNSS outages, the following can be observed:

(1). For the horizontal position error (i.e., in the North and East directions), the proposed GNSS/INS/LiDAR-SLAM integrated navigation system was the smallest, and the GNSS/INS integrated navigation system was the largest. The vertical position error of the proposed GNSS/INS/LiDAR-SLAM integrated navigation system was similar to the GNSS/INS integrated navigation system, and Cartographer had the largest vertical position error. In the sixth outage when the vehicle was at rest, the Cartographer and GNSS/INS/LiDAR-SLAM integrated navigation system significantly reduced the horizontal positioning drift with the aid of the LiDAR-SLAM compared to the GNSS/INS integrated navigation system.

(2). Cartographer had the largest attitude error, especially in the roll and pitch, which is because Cartographer assumes low dynamic motion of the vehicle and uses gravity to solve the horizontal attitude (similar to the inclinometer principle). The attitude error of the proposed GNSS/INS/LiDAR-SLAM integrated navigation system was equivalent to that of the GNSS/INS integrated navigation system.

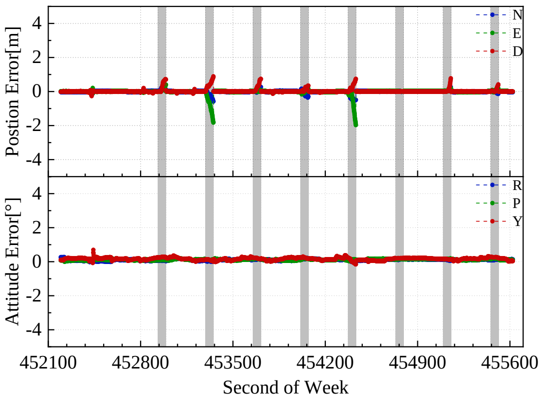

Figure 9. Position and attitude errors of the GNSS/INS/LiDAR-SLAM integrated navigation system in the GNSS outage simulation test #2.

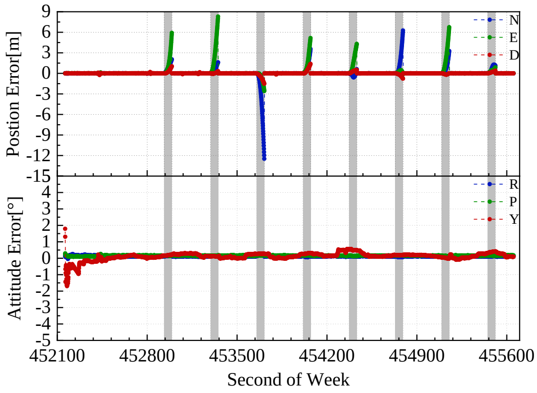

Figure 10. Position and attitude errors of the GNSS/INS integrated navigation system in the GNSS outage simulation test #2.

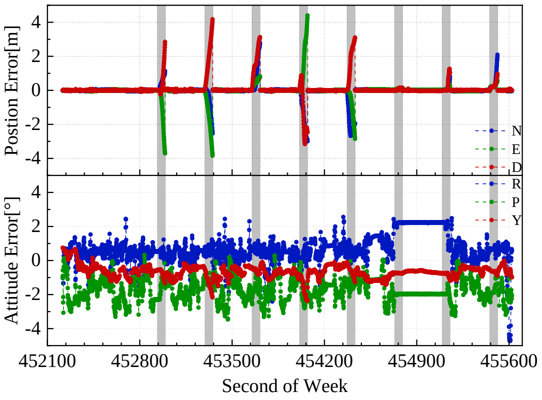

Figure 11. Position and attitude errors of Cartographer in the GNSS outage simulation test #2.

In the three open-sky tests, a total of 26 GNSS outages were simulated. Based on the error statistics in Table 2, the following information can be obtained by comparison with the GNSS/INS integrated navigation system:

(a) The position error RMS in the North, East and vertical directions of the proposed GNSS/INS/LiDAR-SLAM navigation system was reduced by 82.2%, 79.6%, and 17.2%, respectively. The Cartographer's North and East position error RMS was reduced by 47.4% and 44.8%, respectively, but the vertical position error RMS is degraded by 2.3 times.

(b) The aid of the LiDAR-SLAM did not significantly improve the attitude accuracy in the GNSS/INS/LiDAR-SLAM integrated navigation system. The RMS values of the Cartographer's roll, pitch and yaw errors were increased by 9.5, 13.7, and 7.2 times, respectively, because of the dedicated algorithm design for low speed and smooth motion, which does not fit the test cases in this paper.

Table 2. Errors statistics for the GNSS outage simulation tests.

		Position Error [m]			**Attitude Error [°]**		
		N	**E**	**D**	**R**	**P**	**Y**
Proposed GNSS/INS/LiDAR-SLAM	RMS	0.943	1.114	0.721	0.151	0.182	0.213
	MAX	3.212	3.161	1.112	0.234	0.253	0.323
GNSS/INS	RMS	5.298	5.469	0.871	0.136	0.175	0.225
	MAX	12.475	11.328	2.185	0.197	0.217	0.562
Cartographer	RMS	2.789	3.019	2.898	1.438	2.589	1.865
	MAX	4.835	4.439	4.169	2.320	3.602	2.896

The relative position errors (i.e., percentage of position error over travel distance) in the GNSS outages are shown in Figure 12. The left y axis corresponding to the dotted lines in the figure is the relative position error of the three methods, and the right y axis corresponding to the strip is the travel distance in the GNSS outage. In the 16th outage (that is, the sixth outage in test #2, as shown in Figures 9–11), the vehicle was at rest, and the relative position error was not calculated. The average relative position errors of the GNSS/INS/LiDAR-SLAM integrated navigation system, GNSS/INS integrated navigation system and Cartographer were 0.26%, 1.46%, and 0.92% of the travel distance, respectively. The aid of LiDAR-SLAM effectively reduces the relative position error in the GNSS outages.

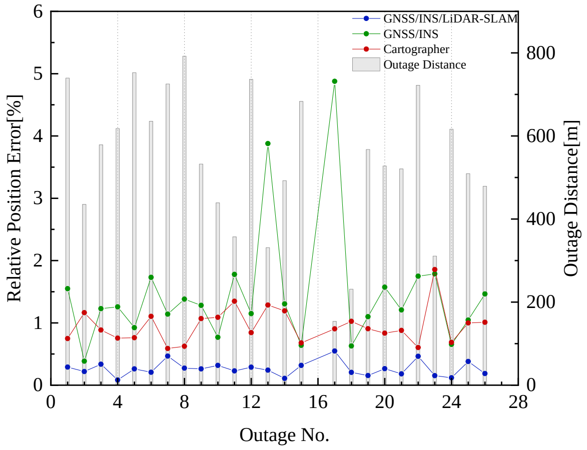

Figure 12. Relative position errors in the GNSS outage simulation tests.

Based on the GNSS outage simulation tests, the aid of LiDAR-SLAM can effectively reduce the position accuracy when the GNSS is unavailable. The horizontal position accuracy of the GNSS/INS/LiDAR-SLAM integrated navigation system and Cartographer was better than that of the GNSS/INS integrated navigation system. However, the Cartographer's height and attitude error was greater than that of the GNSS/INS integrated navigation system. The reason is that Cartographer assumes that the vehicle moves at a low speed with a small acceleration; therefore, gravity is used to estimate the horizontal attitude, and the modelling for the bias of the IMU is ignored. Thus, during the tests in this paper, the horizontal attitude error and the vertical position error of Cartographer were relatively large.

In addition to the open-sky tests with simulated GNSS outage results, the trajectories of the three methods and the reference truth in the urban area test are shown in Figure 13. According to

Figure 8, there were approximately 150 s of poor satellite signals in the test, and the GNSS positioning quality is generally unstable. The position and attitude errors are shown in Figures 14–16, and the statistics for these errors are shown in Table 3. Similar to the three GNSS outage simulation tests, the GNSS/INS/LiDAR-SLAM integrated navigation system had the best positional accuracy. Because of the poor quality of GNSS positioning for a long time, the GNSS/INS integrated navigation system had a large yaw drift error in addition to the large position error compared to the GNSS/INS/LiDAR-SLAM integrated navigation system and Cartographer. Therefore, the aid of LiDAR-SLAM can also improve the yaw accuracy for very tough scenarios.

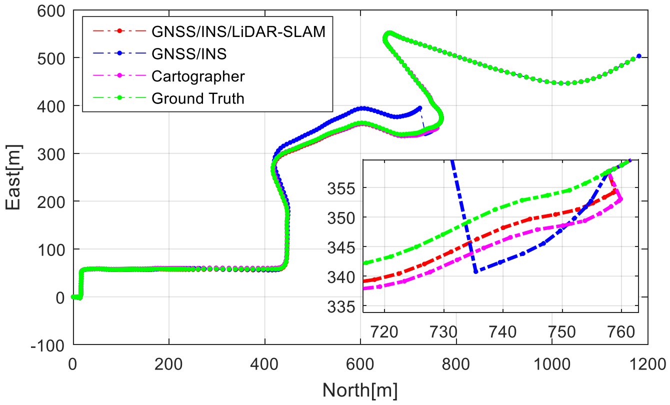

Figure 13. Trajectories in the urban area test.

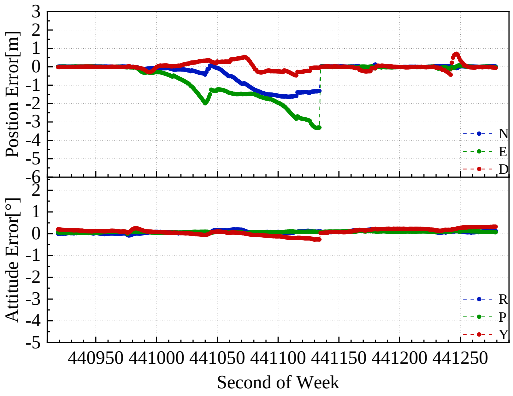

Figure 14. Position and attitude errors of the GNSS/INS/LiDAR-SLAM integrated navigation system in the urban area test.

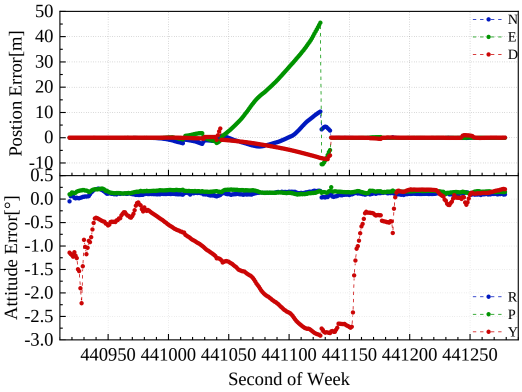

Figure 15. Position and attitude errors of the GNSS/INS integrated navigation system in the urban area test.

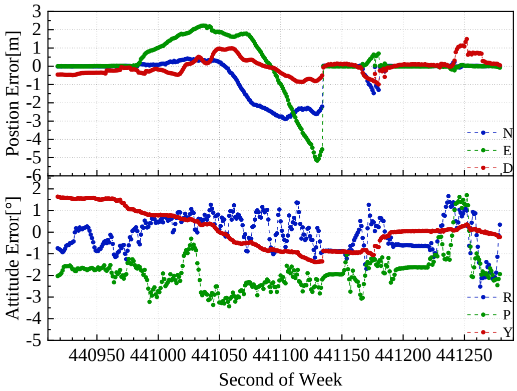

Figure 16. Position and attitude errors of Cartographer in the urban area test.

Table 3. Error statistics for the urban area test.

		Position Error [m]			**Attitude Error [°]**		
		N	**E**	**D**	**R**	**P**	**Y**
Proposed GNSS/INS/LiDAR-SLAM	RMS	0.615	1.105	0.198	0.099	0.083	0.166
	MAX	1.635	3.328	0.707	0.215	0.142	0.323
GNSS/INS	RMS	2.031	11.757	2.389	0.116	0.160	1.375
	MAX	10.338	45.485	8.499	0.223	0.253	2.910
Cartographer	RMS	1.051	1.395	0.463	0.780	2.042	0.867
	MAX	2.891	5.166	1.488	2.517	3.427	1.637

A solution of GNSS/LiDAR/INS navigation is demonstrated in this paper and verified by the real-world tests. There are not too much complicated theoretical conclusions, such as the observability, which should be validated by controllable simulation data have been proposed in this paper. And because in the simple and controllable simulated environment, error models and parameters are clear, the algorithms always get a good result than in real scenarios and it's not persuasive. Consequently, to indicate the performance of this solution, the data collected in real-world scenarios is convincing and this is the only data source in this paper. In addition, the system in this paper requires the raw LiDRA data, which is not easy to simulate.

The results of the four vehicle tests (three open-sky and one urban street) have shown that the position and attitude accuracy of the proposed GNSS/INS/LiDAR-SLAM integrated navigation system is the best, especially in a weak or blocked GNSS signal environment. Cartographer has the largest roll, pitch and elevation errors because the motion model of the algorithm is only suitable for low-speed motion. Due to the lack of aiding information, the GNSS/INS integrated navigation system gradually degrades the position and attitude when GNSS is not available, and LiDAR-SLAM assistance can play a significant role in maintaining navigation accuracy.

5. Conclusions

In this paper, a GNSS/INS/LiDAR-SLAM integrated navigation system is proposed. The MEMS-IMU mechanization is applied for pose estimation. Through graph optimization, the GNSS position, IMU preintegration results and the relative pose obtained from LiDAR scan matching are fused. In addition, the use of a sliding window ensures that the computational load of the graph optimization does not increase with time. The vehicle tests show that the GNSS/INS/LiDAR-SLAM integrated navigation system can effectively reduce the position errors in the horizontal directions during the GNSS outage periods compared with the GNSS/INS integrated navigation system. For the position error in the vertical direction and the attitude error, the two systems perform similarly. In addition, compared with the benchmark method of GNSS/INS/LiDAR-Google Cartographer, the use of the IMU information in the proposed algorithm is more reasonable and sufficient, thus improving both the position and attitude accuracies. Finally, the relative position error of the proposed GNSS/INS/LiDAR-SLAM method during the GNSS outage was reduced from 1.46% (GNSS/INS) to 0.92% (Cartographer) to 0.26% of the travel distance.

For future works, the GNSS/INS/LiDAR integrated navigation system in this paper cannot yet eliminate the dynamic objects from the environment. Therefore, it is necessary to add the recognition and elimination mechanism of dynamic objects in the environment. Furthermore, a procreated background map may be added to improve the robustness and navigation positioning accuracy of the integrated navigation system, such as for applications for autodrive and mobile robots.

Author Contributions: L.C. and X.N. conceived and designed the experiments, wrote the paper, and performed the experiments; T.L. performed the land vehicle tests and provided the LiDAR, IMU, and GNSS raw data; and J.T. and C.Q. reviewed the paper and gave constructive advice.

Funding: This research was funded by the "National Key Research and Development Program" (No. 2016YFB0501803 and No. 2016YFB0502202) and the "Fundamental Research Funds for the Central Universities" (No. 2042018KF0242).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Shin, E.-H. *Estimation Techniques for Low-Cost Inertial Navigation*; The University of Calgary: Calgary, Canada, 2005.
2. Pagliaria, D.; Pinto, L.; Reguzzoni, M.; Rossi, L. Integration of kinect and low-cost gnss for outdoor navigation. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2016**, *XLI-B5*, 565–572. [CrossRef]
3. Rossi, L.; De Gaetani, C.; Pagliari, D.; Realini, E.; Reguzzoni, M.; Pinto, L. Comparison between rgb and rgb-d cameras for supporting low-cost gnss urban navigation. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2018**, *42*, 991–998. [CrossRef]
4. Cadena, C.; Carlone, L.; Carrillo, H.; Latif, Y.; Scaramuzza, D.; Neira, J.; Reid, I.; Leonard, J.J. Past, Present, and Future of Simultaneous Localization and Mapping: Toward the Robust-Perception Age. *IEEE Trans. Robot.* **2016**, *32*, 1309–1332. [CrossRef]
5. Walter, M.R.; Eustice, R.M.; Leonard, J.J. Exactly Sparse Extended Information Filters for Feature-based SLAM. *Int. J. Robot. Res.* **2007**, *26*, 335–359. [CrossRef]
6. Deschaud, J.-E. IMLS-SLAM: Scan-to-model matching based on 3D data. In Proceedings of the 2018 IEEE International Conference on Robotics and Automation (ICRA2018), Brisbane, QL, Australia, 21–25 May 2018; pp. 2480–2485.
7. Behley, J.; Stachniss, C. Efficient Surfel-based Slam Using 3d Laser Range Data in Urban Environments. Available online: http://roboticsproceedings.org/rss14/p16.pdf (accessed on 25 April 2019).
8. Zhang, J.; Singh, S. LOAM: Lidar Odometry and Mapping in Real-time. In Proceedings of the Robotics: Science and Systems, Berkeley, CA, USA, 14–16 July 2014; 2014.
9. Qin, T.; Li, P.; Shen, S. VINS-Mono: A Robust and Versatile Monocular Visual-Inertial State Estimator. *IEEE Trans. Robot.* **2018**, *34*, 1004–1020. [CrossRef]
10. Zhang, J.; Singh, S. Visual-lidar odometry and mapping: Low-drift, robust, and fast. In Proceedings of the 2015 IEEE International Conference on Robotics and Automation (ICRA2015), Seattle, WA, USA, 26–30 May 2015; pp. 2174–2181.
11. Martínez, J.L.; González, J.; Morales, J.; Mandow, A.; García-Cerezo, A.J. Mobile robot motion estimation by 2D scan matching with genetic and iterative closest point algorithms. *J. Field Robot.* **2006**, *23*, 21–34. [CrossRef]
12. Censi, A. An ICP variant using a point-to-line metric. In Proceedings of the 2008 IEEE International Conference on Robotics and Automation, Pasadena, CA, USA, 19–23 May 2008.
13. Aleksandr, S.; Haehnel, D.; Thrun, S. Generalized-ICP. In Proceedings of the Robotics: Science and Systems, Seattle, WA, USA, 28 June–1 July 2009; p. 435.
14. Burguera, A.; González, Y.; Oliver, G. On the use of likelihood fields to perform sonar scan matching localization. *Auton. Robot.* **2009**, *26*, 203–222. [CrossRef]

15. Hess, W.; Kohler, D.; Rapp, H.; Andor, D. Real-time loop closure in 2D LIDAR SLAM. In Proceedings of the IEEE International Conference on Robotics and Automation, Stockholm, Sweden, 16–21 May 2016; pp. 1271–1278.
16. Gao, Y.; Liu, S.; Atia, M.M.; Noureldin, A. INS/GPS/LiDAR Integrated Navigation System for Urban and Indoor Environments Using Hybrid Scan Matching Algorithm. *Sensors* **2015**, *15*, 23286–23302. [CrossRef] [PubMed]
17. Durrant-Whyte, H.; Bailey, T. Simultaneous localization and mapping: Part I. *IEEE Robot. Autom. Mag.* **2006**, *13*, 99–110. [CrossRef]
18. Qian, C.; Liu, H.; Tang, J.; Chen, Y.; Kaartinen, H.; Kukko, A.; Zhu, L.; Liang, X.; Chen, L.; Hyyppä, J. An integrated GNSS/INS/LiDAR-SLAM positioning method for highly accurate forest stem mapping. *Remote Sens.* **2016**, *9*, 3. [CrossRef]
19. Shamsudin, A.U.; Ohno, K.; Hamada, R.; Kojima, S.; Westfechtel, T.; Suzuki, T.; Okada, Y.; Tadokoro, S.; Fujita, J.; Amano, H. Consistent map building in petrochemical complexes for firefighter robots using SLAM based on GPS and LIDAR. *Robomech. J.* **2018**, *5*, 7. [CrossRef]
20. Kukko, A.; Kaijaluoto, R.; Kaartinen, H.; Lehtola, V.V.; Jaakkola, A.; Hyyppä, J. Graph SLAM correction for single scanner MLS forest data under boreal forest canopy. *Isprs J. Photogramm. Remote Sens.* **2017**, *132*, 199–209. [CrossRef]
21. Pierzchała, M.; Giguère, P.; Astrup, R. Mapping forests using an unmanned ground vehicle with 3D LiDAR and graph-SLAM. *Comput. Electron. Agric.* **2018**, *145*, 217–225. [CrossRef]
22. Cartographer. Latest. Available online: https://google-cartographer.readthedocs.io/en/latest/ (accessed on 9 July 2018).
23. Forster, C.; Carlone, L.; Dellaert, F.; Scaramuzza, D. On-Manifold Preintegration for Real-Time Visual–Inertial Odometry. *IEEE Trans. Robot.* **2015**, *33*, 1–21. [CrossRef]
24. Yahao, C. *Algorithm Implementation of Pseudorange Differential BDS/INS Tightly-Coupled Integration and GNSS Data Quality Control Analysis*; Wuhan University: Wuhan, China, 2013.
25. Jun, W.; Gongmin, Y. *Strapdown Inertial Navigation Algorithm and Integrated Navigation Principles*; Northwestern Polytechnical University Press Co. Ltd.: Xi'an, China, 2016.
26. Dehai, Z. *Point Cloud Library PCL Learning Tutorial*; Beihang University Press: Beijing, China, 2012.
27. Slerp. Available online: https://en.wikipedia.org/wiki/Slerp (accessed on 2 October 2018).
28. Olson, E.B. Real-time correlative scan matching. In Proceedings of the IEEE International Conference on Robotics and Automation, Kobe, Japan, 12–17 May 2009; pp. 1233–1239.
29. *Ceres Solver*, 1.14. Available online: http://ceres-solver.org (accessed on 23 March 2018).
30. Lekien, F.; Marsden, J.E. Tricubic Interpolation in Three Dimensions. *Int. J. Numer. Methods Eng.* **2010**, *63*, 455–471. [CrossRef]
31. Clausen, J. Branch and Bound Algorithms—Principles and Examples. *Parallel Process. Lett.* **1999**, *22*, 658–663.
32. Xiangyuan, K.; Jiming, G.; Zongquan, L. *Foundation of Geodesy*, 2nd ed.; Wuhan University Press: Wuhan, China, 2010.
33. Sola, J. Quaternion kinematics for the error-state Kalman filter. *arXiv* **2017**, arXiv:1711.02508.
34. Yongyuan, Q. *Kalman Filter and Integrated Navigation Principle*; Northwestern Polytechnical University Press Co. Ltd.: Xi'an, China, 2000.
35. Sibley, G.; Matthies, L.; Sukhatme, G. Sliding window filter with application to planetary landing. *J. Field Robot.* **2010**, *27*, 587–608. [CrossRef]
36. Zhang, F. *The Schur Complement and Its Applications*; Springer: New York, NY, USA, 2005.
37. Eckenhoff, K.; Paull, L.; Huang, G. Decoupled, consistent node removal and edge sparsification for graph-based SLAM. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Daejeon, Korea, 9–14 October 2016; pp. 3275–3282.