sensors

MDPI

Article

Pole-Like Object Extraction and Pole-Aided GNSS/IMU/LiDAR-SLAM System in Urban Area

Tianyi Liu [1], Le Chang [1], Xiaoji Niu [1,2,3] and Jingnan Liu [1,2,3,*]

1. GNSS Research Center, Wuhan University, 129 Luoyu Road, Wuhan 430079, China; liutianyi@whu.edu.cn (T.L.); changlesgg@whu.edu.cn (L.C.); xjniu@whu.edu.cn (X.N.)
2. Artificial Intelligence Institute, Wuhan University, 129 Luoyu Road, Wuhan 430079, China
3. Collaborative Innovation Center of Geospatial Technology, Wuhan University, 129 Luoyu Road, Wuhan 430079, China

* Correspondence: jnliu@whu.edu.cn

Received: 28 October 2020; Accepted: 9 December 2020; Published: 13 December 2020

Abstract: Vision-based sensors such as LiDAR (Light Detection and Ranging) are adopted in the SLAM (Simultaneous Localization and Mapping) system. In the 16-beam LiDAR aided SLAM system, due to the difficulty of object detection by sparse laser data, neither the grid-based nor feature point-based solution can avoid the interference of moving objects. In an urban environment, the pole-like objects are common, invariant and have distinguishing characteristics. Therefore, it is suitable to bring more robust and reliable positioning results as auxiliary information in the process of vehicle positioning and navigation. In this work, we proposed a scheme of a SLAM system using a GNSS (Global Navigation Satellite System), IMU (Inertial Measurement Unit) and LiDAR sensor using the position of pole-like objects as the features for SLAM. The scheme combines a traditional preprocessing method and a small scale artificial neural network to extract the pole-like objects in environment. Firstly, the threshold-based method is used to extract the pole-like object candidates from the point cloud, and then, the neural network is applied for training and inference to obtain pole-like objects. The result shows that the accuracy and recall rate are sufficient to provide stable observation for the following SLAM process. After extracting the poles from the LiDAR point cloud, their coordinates are added to the feature map, and the nonlinear optimization of the front end is carried out by utilizing the distance constraints corresponding to the pole coordinates; then, the heading angle and horizontal plane translation are estimated. The ground feature points are used to enhance the elevation, pitch and roll angle accuracy. The performance of the proposed navigation system is evaluated through field experiments by checking the position drift and attitude errors during multiple two-min mimic GNSS outages without additional IMU motion constrain such as NHC (Nonholonomic Constrain). The experimental results show that the performance of the proposed scheme is superior to that of the conventional feature point grid-based SLAM with the same back end, especially in congested crossroads where slow-moving vehicles are surrounded and pole-like objects are rich in the environment. The mean plane position error during two-min GNSS outages was reduced by 38.5%, and the root mean square error was reduced by 35.3%. Therefore, the proposed pole-like feature-based GNSS/IMU/LiDAR SLAM system can fuse condensed information from those sensors effectively to mitigate positioning and orientation errors, even in a short-time GNSS denied environment.

Keywords: urban environment; pole extraction; neural network; LiDAR; integrated navigation

1. Introduction

An automatic driving system is an unremitting pursuit of intelligent vehicles, and the navigation ability is the essential guarantee for vehicles to successfully complete the automatic driving task safely and accurately. A Global Navigation Satellite System (GNSS), which includes a GPS (Global Positioning System), BDS (BeiDou Navigation Satellite System) [1,2], GLONASS and Galileo, can easily provide the position information of the target for their users. However, in urban environment, due to the existence of buildings and viaducts, low-quality observations, i.e., multi-path signals and NLOS (Non-Line-Of-Sight) signals, are frequently obtained, causing unreliable GNSS positioning results. Although some processes [3] can be absorbed in the navigation solution to improve the positioning, the precision is still insufficient for automatic driving. Therefore, it is difficult to ensure the accuracy of vehicle positioning. In the widely used integrated navigation scheme, data from GNSS and Inertial Measurement Unit (IMU) are fused by a Kalman filter [4] or graph optimization [5–7]. However, errors from medium-end and low-end IMU will accumulate rapidly over time [8,9], and the positioning error will also appear after the GNSS measurement quality deteriorates for a period of time. Therefore, the fusion of multisource sensors is the inevitable choice of an automatic driving positioning scheme.

Vision information obtained from a camera is widely used in Simultaneous Localization and Mapping (SLAM) [10–15]. Compared to the camera, Light Detection and Ranging (LiDAR) can not only work under different weather and different illumination but, also, get a 3D structure of its surroundings. As an important component of the current automatic driving system or mobile robots, LiDAR can undertake positioning, sensing and mapping tasks, so it has been widely used [16–24]. However, researchers always try to use common and concise features to make it. Zhao and Farrell [25] used the laser reflections of 2D LiDAR scanning to detect lines to associate with mapped building planes and then used them as anchors to get the position and orientation of the LiDAR. Im et al. [26] used point cloud data to extract the corner features of buildings to locate vehicles in urban environments. In the process of urban modernization, for the sake of the attractiveness of buildings, many of them in commercial and residential areas use a large amount of glass materials as their walls. A glass material will produce specular reflection and irregular scattering to the electromagnetic wave emitted by LiDAR, which brings remarkable LiDAR measurement errors. In congested areas, moving vehicles such as vans, trucks and buses always provide unreliable surface or edge feature points for LiDAR scan matching. As a result of this, errors of position estimations and orientation determinations are brought about in dynamic environments. However, in urban construction, the infrastructure of traffic poles, streetlight poles and tree planting often comply with certain standards and remain unchanged for a long time. Therefore, in the LiDAR integrated navigation scheme, it is advantageous to carry out short-term unmanned ground vehicle navigation positioning or maintaining a long-term high-definition map by using pole-like objects as a general feature in an urban environment. In Schaefer et al. [27] and Weng et al. [28], the authors use the navigation system with LiDAR to generate the map in advance and then extract the pole-like objects in the map as landmarks. Finally, the effectiveness of global positioning by matching current scanning to extracted landmarks was proven. However, when the vehicle is in an unknown environment or on a rebuilt street, it is impossible to obtain the prebuild map for global localization. In Cabo et al. [29], Rodríguez-Cuenca et al. [30], Yu et al. [31], Zheng et al. [32] and Wu et al. [33], dense and high-ranging precision LiDAR data obtained from a MLS (Mobile Laser Scanner) were used to extract street furniture in a city, and excellent results were shown in these papers. In these survey tasks above, platform vehicles can move slowly, and object extraction can be done in postprocessing. However, in an automatic driving task, only low-cost sensors can be accepted in the system, and the designed algorithms should also work in high-speed driving scenarios in real time. As a consequence, the ability of pole extraction in a real time point and SLAM is necessary for autonomous vehicles. Semantic tasks such as classification, segmentation and the object detection of a point cloud have been researched for years. Qi et al. [34,35] proposed deep-learning methods for point cloud classification. Wu et al. referred to FCN (Full Convolutional Networks) [36]—the segmentation task in the image processing field and the designed Squeezeseg network and its second version [37,38]—to tackle the segmentation problem on a relative dense point cloud. Zhou et al. [39] came up with an idea of using an end-to-end network to detect vehicles, pedestrians and cyclists in a KITTI (Karlsruhe Institute of Technology and Toyota Technological Institute) dataset. Although they have good performance on these tasks, all of them were designed for dense point cloud data, and GPUs are necessary in these algorithms, alongside high-power consumption. To make full use of power in vehicles and obtain navigation results in time, a slim but efficient algorithm should be proposed to extract pole-like objects from the sparse point cloud obtained by less-beam LiDAR like Velodyne VLP-16.

In this paper, we propose and evaluate the navigation and global localization scheme of automatic driving vehicles by using the pole-like objects in an urban environment, such as tree trunks, light poles and traffic poles, as auxiliary information without using a prebuild map. The scheme consists of a pole extraction algorithm by combining a tradition threshold-based method with a slim artificial neural network to consider both the efficiency and accuracy. The sensors of the proposed system include GNSS, IMU and a 16-beam LiDAR.

2. Methods

2.1. Coordinate Definition

Since three sensors are used in the proposed system, the following several coordinate systems are defined, and all systems follow the right-hand rule:

1 Body coordinate system (b-frame): The coordinate system of the IMU, where the X-axis is pointing right, Y-axis is pointing forward and Z-axis is pointing down.
2 LiDAR coordinate system (l-frame): This coordinate system is defined as "l", with the X-axis, Y-axis and Z-axis pointing right, forwards and up, respectively.
3 World coordinate frame (w-frame): The coordinate system of the GNSS positioning results, with the initial GNSS position as the origin, the X-axis pointing east, the Y-axis pointing north and the Z-axis pointing up.
4 Map coordinate system (m-frame): Its origin is the position where the SLAM is initialized, and the X-Y plane is the local horizontal plane. The X-axis of the m-frame is parallel to the X-axis of the b-frame at the time when the system is initialized.

2.2. Pole Extraction

Pole-like objects are ubiquitous in urban scenarios, and they occur as parts of streetlamps, trees and traffic signs. Due to their invariance under different seasons and weather changes, pole-like objects have become high-quality landmarks for vehicle localization in autonomous driving. To make full use of this reliable and long-lasting feature, pole extraction has aroused great interest in academia, especially in the research field of autonomous vehicle navigation and high-definition map (HD map)-making [40–42]. In Song et al. [43], a point cloud is clustered, and then features such as eigen values, principle components of each cluster, are computed. Then, the feature vector is taken as the input of the back-propagation neural network to get the label of the belonging cluster. However, in reality, traffic signs occlude in the crowns of trees, and tree trunks grow in bushes sometimes. In these cases, the point cloud of traffic signs and tree trunks cannot be segmented into an individual cluster precisely, so the cluster might be classified into an incorrect category, resulting in a low recall of the classification task. Taking navigation and localization into account, a novel method of pole extraction is proposed in this paper. To get high recall of the pole extraction, a traditional method is taken to get pole candidates by relaxed thresholds. Then, features of these candidate point clouds are computed and fed into the neural network, to classify the candidate into pole or non-pole objects.

Considering the sparsity of the point cloud collected from Velodyne PUCK VLP-16 (called VLP-16 in the following paragraphs) in the vertical direction, point clouds are accumulated over 0.2 s to form one frame or a node. The origin of the frame is the origin of LiDAR at the very beginning of each frame. Ground segmentation is taken as the first step of the pole extraction module, and the method in LeGO-LOAM [44] is adopted in this stage. Then, nonground points are projected onto the X-Y plane and then rasterized into regular 2D grids. A connected-component-labeling (CCL) algorithm is applied to get each individual cluster of projected nonground point clouds. In order to get pole candidates in each cluster, sliding window searching is used; every time, the corner of each gird is set as the searching center; then, with the given radius, a circle cloud is determined by the given radius. If the number of points in the circle exceeds the certain threshold N_{th} and the height is higher than 1.5 m, the set of points will be regarded as a pole candidate. The whole process is shown in Figure 1.

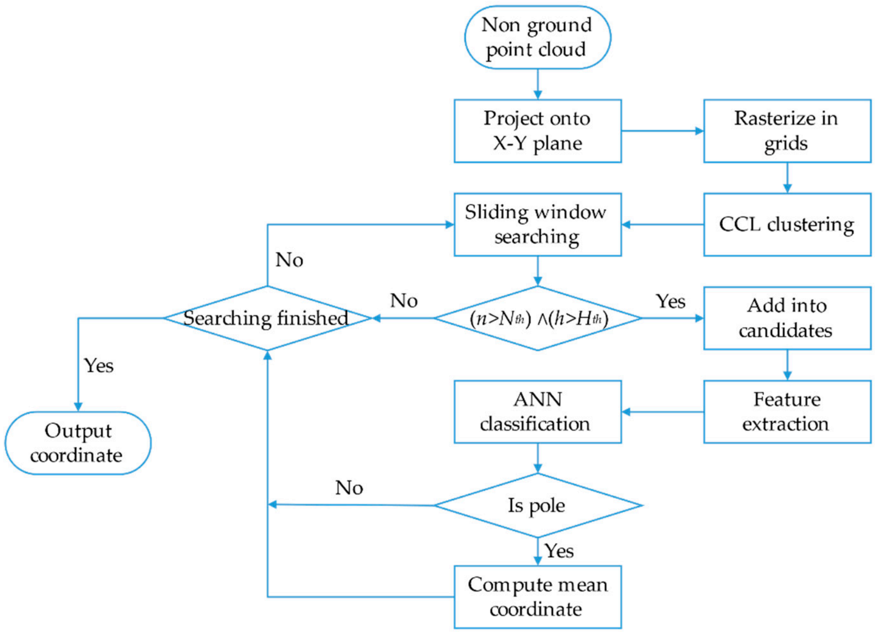


Figure 1. Flow chart of pole candidate extraction from a nonground point cloud. CCL: connected-component-labeling.

Figure 2 shows the process of pole candidate searching: the blue squares are the rasterized grids, the green points are the local point cloud projected on the X-Y plane and the orange circle is the searching field determined by a given radius whose center is the corner of the grid.

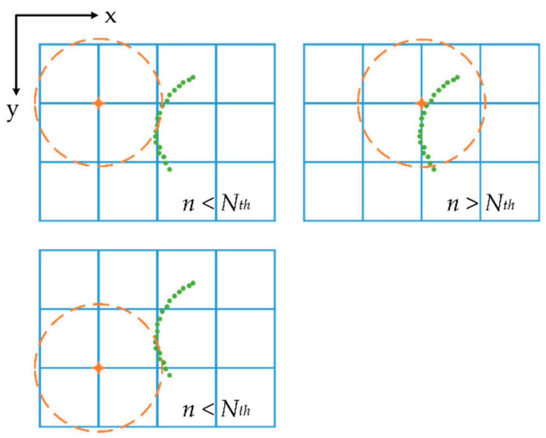


Figure 2. Sliding window searching for pole candidates.

In the previous stages, ground points are removed, and nonground points are clustered into small clusters; therefore, a large number of useless searches are avoided. Multithread techniques can be applied in the sliding window search to ensure the real-time processing of this module.

After pole candidates' extraction, five features of the candidates are computed: the ratio of the number of pole candidates and the number of its neighborhoods, the minimum value in the z-axis, the maximum of the z-axis, variance of the z-axis and mean intensity. Cross entropy loss is employed as the loss function of the designed neural network in pole and non-pole classification tasks. In Equation (1), N is the amount of samples, y_i is the ground truth label of sample i and $\hat{y}_i$ is the probability predicted by the neural network. In the proposed solution, a shallow artificial neural network with two layers is applied as the classifier. Due to the preprocessing of the threshold-based method, both the performance and generalization are good enough for pole extraction in the navigation task.

$$Loss = -\frac{1}{N}\sum_{i=1}^{N}[y_i \cdot \log(\hat{y}_i) + (1-y_i)\cdot\log(1-\hat{y}_i)], \tag{1}$$

2.3. *Integrated Navigation Solution*

The whole system can be subdivided into the following parts: ground segmentation, pole extraction, pose extrapolation, scan-to-map matching, IMU preintegration and pose graph optimization. The flow chart is shown in Figure 3.

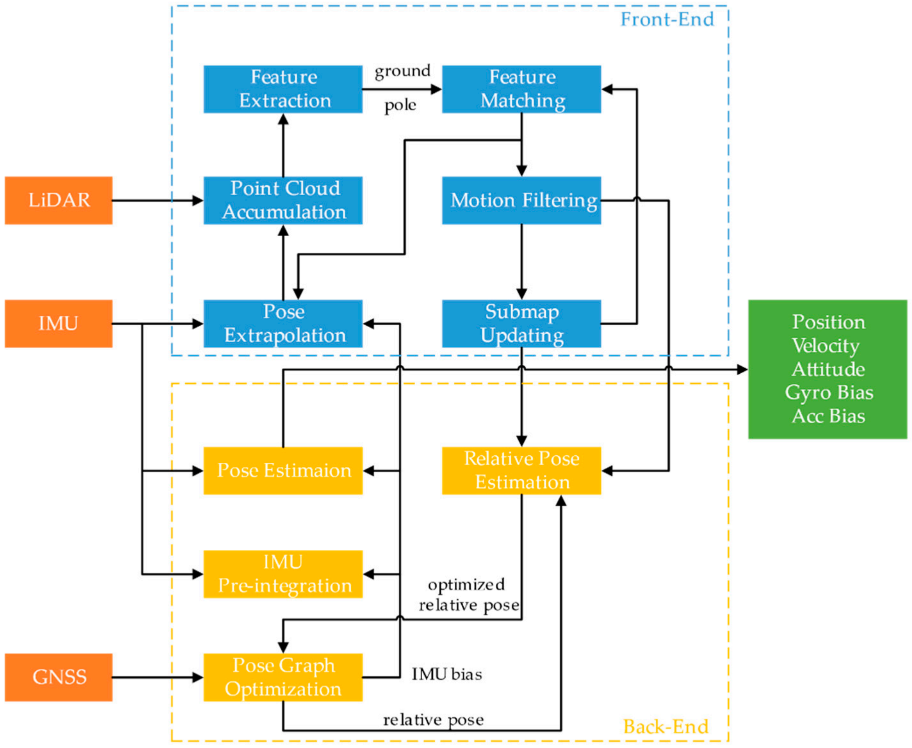


Figure 3. Flow chart of the proposed integrated navigation system. LiDAR: Light Detection and Ranging, IMU: Inertial Measurement Unit and GNSS: Global Navigation Satellite System.

2.3.1. Pose Extrapolation

The IMU here in this system is employed to do pose extrapolation for LiDAR motion compensation and provide an initial guess of the transformation of the sensors. Equations (2)–(5) show the process of pose extrapolation by the output of the IMU.

$$\Delta\tau_i = \tau_i - \tau_{i-1}, \tag{2}$$

$$\mathbf{v}^m_{mb_i} = \mathbf{v}^m_{mb_{i-1}} + \mathbf{C}^m_{b_{i-1}}\Delta\mathbf{v}^{b_{i-1}}_{b_{i-1}b_i} + \mathbf{g}^m\Delta\tau_i, \tag{3}$$

$$\mathbf{P}^m_{mb_i} = \mathbf{P}^m_{mb_{i-1}} + \mathbf{v}^m_{mb_{i-1}}\Delta\tau_i + \frac{1}{2}\mathbf{g}^m\Delta{\tau_i}^2 + \frac{1}{2}\mathbf{C}^m_{b_{i-1}}\mathbf{v}^{b_{i-1}}_{b_{i-1}b_i}\Delta\tau_i, \tag{4}$$

$$\mathbf{q}^m_{b_i} = \mathbf{q}^m_{b_{i-1}} \otimes \mathbf{q}^{b_{i-1}}_{b_i}. \tag{5}$$

τ_i in Equation (2) is the ith timestamp of the sensor; b_i corresponds to the b-frame at time i; $v^m_{mb_i}$ is the velocity in the b-frame relative to the m-frame, which projected on the m-frame at time i; $\mathbf{C}^m_{b_{i-1}}$ is the direction cosine matrix representing the rotation from the b-frame to m-frame; $\mathbf{g}^m$ is denoted as the gravity in the m-frame; **P**, **v** and **q** are the position, velocity and pose (represented as the quaternion), respectively and the symbol $\otimes$ is the multiplication of the quaternion.

2.3.2. Feature Matching

In this paper, laser points reflected from ground are assumed to fit well by the same set of plane parameters locally. This supposedly was validated by Shan et al. [44] and Liu et al. [45]. The method of ground segmentation mentioned in Himmelsbach et al. [46] is applied to extract plane feature points on the ground. Next, the voxel filter is applied to reduce the computational cost in the feature-matching process. Note the transformation matrix from the l-frame to the m-frame as $\mathbf{T}^m_l$ composed by the rotation matrix and translation vector and can be written as Equation (6) in a block matrix format:

$$\mathbf{T}^m_l = \begin{bmatrix} \mathbf{C}^m_l & \mathbf{t}^m_l \\ 0_{(1\times3)} & 1 \end{bmatrix}. \tag{6}$$

The inverse matrix of $\mathbf{T}^m_l$ is $\left(\mathbf{T}^m_l\right)^{-1}$, which satisfies

$$\left(\mathbf{T}^m_l\right)^{-1} = \mathbf{T}^l_m = \begin{bmatrix} \left(\mathbf{C}^m_l\right)^{\mathrm{T}} & -\left(\mathbf{C}^m_l\right)^{\mathrm{T}} \cdot \mathbf{t}^m_l \\ 0_{(1\times3)} & 1 \end{bmatrix}. \tag{7}$$

The optimized transformation matrix from the l-frame to the m-frame at timestamp $i-1$ is $\mathbf{T}^m_{l,i-1}$, and the lever arm and misalign angle between the IMU and LiDAR are denoted as $\mathbf{t}^l_b$ and $\mathbf{C}^l_b$. The transformation $\mathbf{T}^{b,i-1}_{b,i}$ from time i to $i-1$ can be calculated by the output of the IMU. Combined with $\mathbf{t}^l_b$ and $\mathbf{C}^l_b$ mentioned above, the initial guess of the transformation of the l-frame from time i to time $i-1$, $\hat{\mathbf{T}}^{l,i-1}_{l,i}$ can be computed as

$$\hat{\mathbf{T}}^{l,i-1}_{l,i} = \mathbf{T}^{l,i}_b \cdot \hat{\mathbf{T}}^{b,i-1}_{b,i} \cdot \mathbf{T}^b_{l,i-1} = \begin{bmatrix} \mathbf{C}^l_b & \mathbf{t}^l_b \\ 0_{(1\times3)} & 1 \end{bmatrix} \cdot \hat{\mathbf{T}}^{b,i-1}_{b,i} \cdot \begin{bmatrix} \left(\mathbf{C}^l_b\right)^{\mathrm{T}} & -\left(\mathbf{C}^l_b\right)^{\mathrm{T}} \cdot \mathbf{t}^l_b \\ 0_{(1\times3)} & 1 \end{bmatrix}, \tag{8}$$

Then, at time i, the transformation from the l-frame to the m-frame is obtained:

$$\hat{\mathbf{T}}^m_{l,i} = \mathbf{T}^m_{l,i-1} \cdot \hat{\mathbf{T}}^{l,i-1}_{l,i}. \tag{9}$$

The laser reflection of LiDAR is denoted as

$$\mathbf{p} = \begin{bmatrix} x & y & z \end{bmatrix}^{\mathrm{T}}, \tag{10}$$

while the homogeneous format of the 3D point is

$${}^h\mathbf{p} = \begin{bmatrix} x & y & z & 1 \end{bmatrix}^{\mathrm{T}}. \tag{11}$$

To simplify the Equations, the superscript h is omitted, and the nonhomogeneous format of the points is used in the following equations, where the actually homogeneous coordinate is operated with the 4-dimensional transformation matrix. For the purpose of estimating the roll, pitch angle and elevation, feature points in the l-frame are transformed into the m-frame, and then, 5 neighbor feature points $\mathbf{p}^{m,j}, j \in \{1,2,3,4,5\}$ are found in the m-frame to fit a plane. The formula of the plane is

$$Ax + By + Cz + d = 0, \tag{12}$$

where $\sqrt{A^2+B^2+C^2} = 1$ and unit normal $\mathbf{n} = \begin{bmatrix} A & B & C \end{bmatrix}^{\mathrm{T}}$. Roll, pitch angle and elevation are estimated to minimize the distance between the feature point and fitted plane. However, the feature points too far from the fitted plane would not be absorbed in the cost function. Suppose there are K points that are satisfied with the distance threshold, then the objective function can be written as

$$\underset{\theta^m_{l,roll},\theta^m_{l,pitch},t_z}{\operatorname{argmin}} \sum_{k=1}^{K} \left| \mathbf{n} \cdot \left(\hat{\mathbf{T}}^m_l \cdot \mathbf{p}^{l,k} \right) + d_k \right|. \tag{13}$$

In the above Equation (13), $\theta^m_{l,roll}$, $\theta^m_{l,pitch}$ and t_z are the corresponding components in the transformation matrix representing the roll, pitch angle and height translation, and d_k is the distance between the kth feature point to its local plane.

On the other hand, point clouds of poles are obtained by the algorithm introduced in Section 2.2. Consequently, the mean of x and y marked as pole feature points can be used to optimize the translation in the x and y directions and yaw angle of the vehicle by minimizing the horizontal distance between correspondences in the m-frame:

$$\underset{\theta^m_{l,yaw},t^m_{l,x},t^m_{l,y}}{\operatorname{argmin}} \sum_{k=1}^{K} \|\mathbf{p}^{m,j} - \hat{\mathbf{T}}^m_l \cdot \mathbf{p}^{l,k}\|. \tag{14}$$

2.3.3. Optimization in the Back End

In this paper, the IMU preintegration, GNSS position and relative transformation between the LiDAR nodes and submaps are used as constrains to establish the pose graph for pose optimization in the back end (Chang et al. [47,48]). The variable that is going to be optimized can be written as χ, where $\mathbf{x}_k$ is composed of the position, velocity, attitude, accelerometer bias and gyroscope bias of the IMU at time τ_k. N is the number of nodes, $\mathbf{y}_k$ is the position and attitude of the submap in the m-frame at τ_k and M is the number of submaps:

$$\begin{aligned} \chi &= \left[\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \dots \mathbf{x}_N, \mathbf{y}_1, \mathbf{y}_2, \mathbf{y}_3, \dots \mathbf{y}_M, \mathbf{P}^w_{wm}, \mathbf{q}^w_m\right] \\ \mathbf{x}_k &= \left[\mathbf{P}^m_{mb_k}, \mathbf{v}^m_{mb_k}, \mathbf{q}^m_{mb_k}, \mathbf{b}_{a,k}, \mathbf{b}_{g,k}\right], k \in \{1,2,3,\dots N\} \\ \mathbf{y}_k &= \left[\mathbf{P}^m_{ms_k}, \mathbf{q}^m_{ms_k}\right], k \in \{1,2,3,\dots M\} \end{aligned} \tag{15}$$

The cost functions of pose graph optimization are Equation (16)

$$\underset{\chi}{\operatorname{argmin}} \left\{ \begin{array}{l} \sum\limits_{i\in\alpha} E^2_{GNSS}\left(\mathbf{x}_i, \mathbf{P}^w_{wm}, \mathbf{q}^w_m, \mathbf{p}^w_{g_i}, \mathbf{l}^b_g, \boldsymbol{\sigma}_p\right) + \\ \sum\limits_{\substack{i\in\beta,\\ j\in\kappa}} E^2_{LiDAR}\left(\mathbf{x}_i, \mathbf{y}_j, \mathbf{P}^b_{bl}, \mathbf{q}^l_b, \mathbf{P}^{s_j}_{s_j l_i}, \mathbf{q}^{s_j}_{l_i}, \boldsymbol{\sigma}_{ij}\right) + \\ \sum\limits_{i=1}^{N-1} E^2_{IMU}\left(\mathbf{x}_i, \mathbf{x}_{i+1}, \mathbf{z}^i_{i+1}, \boldsymbol{\sigma}_z\right) \end{array} \right\}. \tag{16}$$

$E^2_{GNSS}(\cdot)$, $E^2_{LiDAR}(\cdot)$ and $E^2_{IMU}(\cdot)$ are the cost functions of the GNSS, LiDAR and IMU, respectively. $\mathbf{p}^w_{g_i}$ is the positioning result in the w-frame, $\mathbf{l}^b_g$ is the lever arm between the GNSS antenna and IMU (pointing to the phase center of the antenna from the centroid of the IMU), $\boldsymbol{\sigma}_p$ is the standard deviation of $\mathbf{p}^w_g$, α is the nodes set with the GNSS positioning result, $\mathbf{P}^{s_j}_{s_j l_i}$ and $\mathbf{q}^{s_j}_{l_i}$ are the relative position and attitude of the node l_i and the submap s_j, $\boldsymbol{\sigma}_{ij}$ is the standard deviation of $\mathbf{P}^{s_j}_{s_j l_i}$ and $\mathbf{q}^{s_j}_{l_i}$, β is the nodes set, κ is the submaps set, $\mathbf{z}^{i}_{i+1}$ is the preintegration result between the t_i node and the t_{i+1} node and $\boldsymbol{\sigma}_z$ is the standard deviation of $\mathbf{z}$. $\mathbf{P}^b_{bl}$ and $\mathbf{q}^l_b$ are the extrinsic parameters of LiDAR and IMU, including the lever arm (pointing to the center of the laser transmitter from the centroid of the IMU) and misaligned angle (projected into the b-frame). According to Chang et al. [47], the cost functions of GNSS and LiDAR can be obtained:

$$
\begin{aligned}
E^2_{GNSS}\left(\mathbf{x}_i, \mathbf{P}^w_{wm}, \mathbf{q}^w_m, \mathbf{p}^w_{g_i}, \mathbf{l}^b_g, \boldsymbol{\sigma}_p\right) &= e\left(\mathbf{x}_i, \mathbf{P}^w_{wm}, \mathbf{q}^w_m, \mathbf{p}^w_{g_i}, \mathbf{l}^b_g\right)^{\mathrm{T}}\left(\boldsymbol{\sigma}_p^2\right)^{-1} e\left(\mathbf{x}_i, \mathbf{P}^w_{wm}, \mathbf{q}^w_m, \mathbf{p}^w_{g_i}, \mathbf{l}^b_g\right) \\
e\left(\mathbf{x}_i, \mathbf{P}^w_{wm}, \mathbf{q}^w_m, \mathbf{p}^w_{g_i}, \mathbf{l}^b_g\right) &= \mathbf{C}^w_m\left(\mathbf{C}^m_{b_i}\mathbf{l}^b_g + \mathbf{p}^m_{mb_i}\right) + \mathbf{P}^w_{wm} - \mathbf{p}^w_{g_i} \\
E^2_{LiDAR}\left(\mathbf{x}_i, \mathbf{y}_j, \mathbf{P}^b_{bl}, \mathbf{q}^b_l, \mathbf{P}^{s_j}_{s_j l_i}, \mathbf{q}^{s_j}_{l_i}, \boldsymbol{\sigma}_{ij}\right) &= e\left(\mathbf{x}_i, \mathbf{y}_j, \mathbf{P}^b_{bl}, \mathbf{q}^b_l, \mathbf{P}^{s_j}_{s_j l_i}, \mathbf{q}^{s_j}_{l_i}\right)^{\mathrm{T}}\left(\boldsymbol{\sigma}_{ij}^2\right)^{-1} e\left(\mathbf{x}_i, \mathbf{y}_j, \mathbf{P}^b_{bl}, \mathbf{q}^b_l, \mathbf{P}^{s_j}_{s_j l_i}, \mathbf{q}^{s_j}_{l_i}\right), \\
e\left(\mathbf{x}_i, \mathbf{y}_j, \mathbf{P}^b_{bl}, \mathbf{q}^b_l, \mathbf{P}^{s_j}_{s_j l_i}, \mathbf{q}^{s_j}_{l_i}\right) &= \begin{bmatrix} \left(\mathbf{C}^w_{s_j}\right)^{-1}\left[\mathbf{p}^m_{mb_i} + \mathbf{C}^w_{b_j}\mathbf{P}^b_{bl} - \mathbf{p}^m_{ms_j}\right] - \mathbf{P}^{s_j}_{s_j l_i} \\ \left[\left(\mathbf{q}^m_{b_i} \otimes \mathbf{q}^b_l\right)^{-1} \otimes \mathbf{q}^m_{s_j} \otimes \mathbf{q}^{s_j}_{l_i}\right]_{xyz} \end{bmatrix} \\
E^2_{IMU}\left(\mathbf{x}_{k-1}, \mathbf{x}_k, \mathbf{z}^{t_{k-1}}_{t_k}, \boldsymbol{\sigma}_z\right) &= e\left(\mathbf{x}_{k-1}, \mathbf{x}_k, \mathbf{z}^{t_{k-1}}_{t_k}\right)^{\mathrm{T}}\left(\boldsymbol{\sigma}_z^2\right)^{-1} e\left(\mathbf{x}_{k-1}, \mathbf{x}_k, \mathbf{z}^{t_{k-1}}_{t_k}\right)
\end{aligned}
\tag{17}
$$

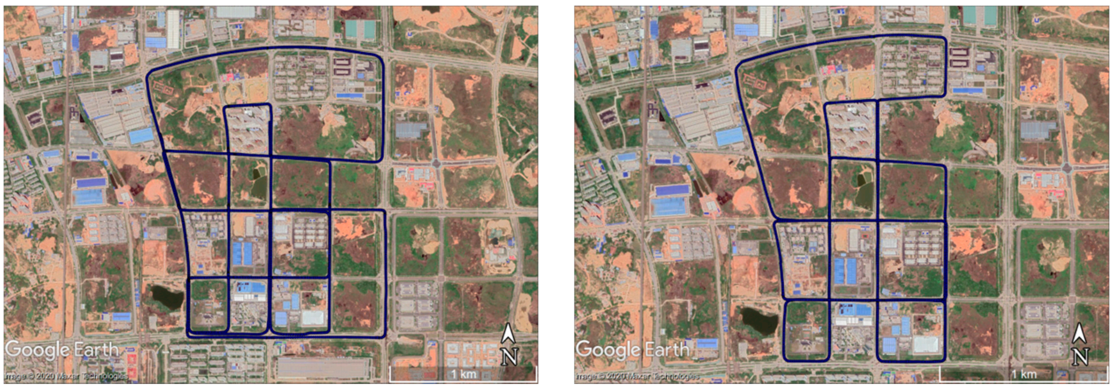
where $e(\cdot)$ is the residual function, and $[\mathbf{q}]_{xyz}$ represents the rotation vector equivalent to $\mathbf{q}$.

$$
\begin{aligned}
&\boldsymbol{\alpha}^{b_{k-1}}_{b_{k-1}} = 0, \boldsymbol{\beta}^{b_{k-1}}_{b_{k-1}} = 0, \mathbf{q}^{b_{k-1}}_{b_{k-1}} = \mathbf{I} \\
&\boldsymbol{\beta}^{b_{k-1}}_{b_i} = \boldsymbol{\beta}^{b_{k-1}}_{b_{i-1}} + \mathbf{C}^{b_{k-1}}_{b_{i-1}}\mathbf{v}^{b_{i-1}}_{b_{i-1}b_i} \\
&\boldsymbol{\alpha}^{b_{k-1}}_{b_i} = \boldsymbol{\alpha}^{b_{k-1}}_{b_{i-1}} + \tfrac{1}{2}\mathbf{C}^{b_{k-1}}_{b_{i-1}}\mathbf{v}^{b_{i-1}}_{b_{i-1}b_i}\Delta\tau_i \\
&\mathbf{q}^{b_{k-1}}_{b_i} = \mathbf{q}^{b_{k-1}}_{b_{i-1}} \otimes \mathbf{q}^{b_{i-1}}_{b_i}
\end{aligned}
\tag{18}
$$

$\boldsymbol{\alpha}^{b_{k-1}}_{b_i}$, $\boldsymbol{\beta}^{b_{k-1}}_{b_i}$ and $\mathbf{q}^{b_{k-1}}_{b_i}$ are the increments of position, velocity and attitude calculated from the IMU preintegration, only related to the output of the IMU, being independent to the initial state. τ_i is some timestamp between $[\tau_{k-1}, \tau_k]$.

3. Experiments

The experimental data was collected in a suburb area in Wuhan City in an open-sky environment to ensure a high-quality GNSS RTK (Real Time Kinematic) signal and the precision of the reference system. The RTK base station was set several kilometers away from the experimental field. The trajectories of the data collection projected on Google Earth are shown in Figure 4.



(**a**) (**b**)

Figure 4. Trajectories of the tests projected on Google Earth: (**a**) trajectory of test #1 and (**b**) trajectory of test #2.

In order to mitigate the correlation between two neighbor outages, the interval between them is set as 180 s, and each outage lasts for 120 s.

The reference system of this experiment consisted of IMU-A15 IMU—a high-level tactical-grade IMU with a laser gyroscope produced by the Leador Spatial Information Technology Corporation (Wuhan, China), a SICK-DFS60E-BECM02048 odometer (SICK, Waldkirch, Germany) and GNSS antenna from NovAtel (Calgary, AB, Canada). The experimental system was equipped with NV-POS1100—a quasi-tactical grade MEMS IMU (Sai MicroElectronics Inc., Beijing, China), Velodyne PUCK VLP-16 (Velodyne Lidar Inc., San Jose, CA, USA) and a common GNSS antenna with a reference system. The IMU-A15 IMU or SICK odometer were not used in the testing system. Figure 5 shows the equipments of the field test.

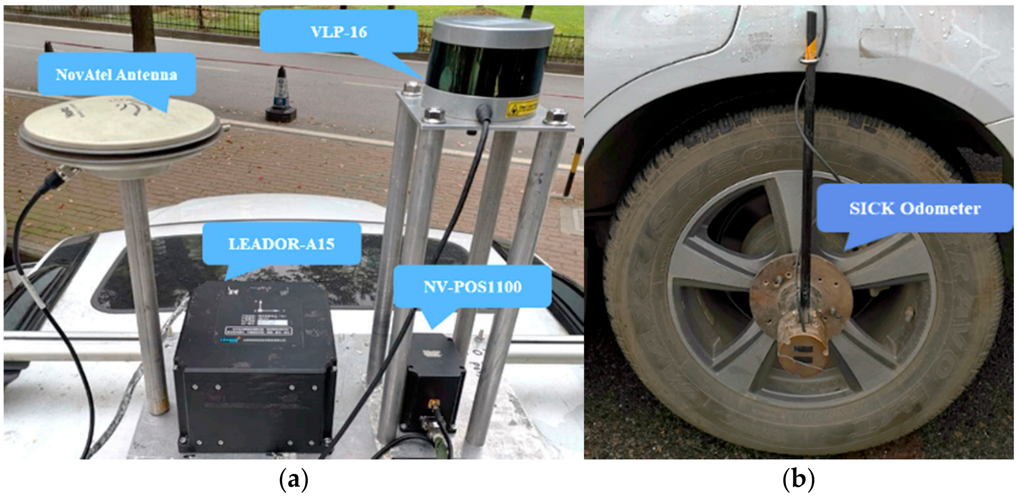


Figure 5. Experimental equipment: (**a**) equipment installed on the roof of the testing vehicle and (**b**) odometer.

The specified parameters of the IMUs are in Table 1.

Table 1. Basic parameters of the Inertial Measurement Unit (IMU)-A15 (reference truth) and NV-POS1100.

IMU	Gyroscope		Accelerometer	
	Bias Instability (°/h)	**Random Walk Noise (°/√h)**	**Bias Instability (mGal)**	**Random Walk Noise (m/s/√h)**
IMU-A15	0.027	0.003	15	0.03
NV-POS1100	10	0.20	1000	0.18

The sampling rate of each sensor is shown in Table 2.

Table 2. Sampling rate of each sensor in the data collection. GNSS: Global Navigation Satellite System.

Sensor	IMU-A15	NV-POS1100	SICK	GNSS Receiver	VLP-16
Sampling Rate	200 Hz	200 Hz	200 Hz	1 Hz	10 Hz

4. Results and Discussion

The performance pole classifications and integrated navigation algorithms will be discussed separately.

In the training stage of the pole candidate classifications, each frame of the point cloud was accumulated by the pose obtained from the front end of Chang et al. [47]; then, the ground was removed, and points were clustered by the method in Section 2.2. Finally, the samples were generated by manual labeling. There were 813 samples in total, with 394 poles and 419 nonpoles. Eighty percent of the samples were utilized as a training set, while the rest of samples were used for testing. To be general, the frames where pole candidates were extracted were expected to cover a diversity of scenarios, including crossroads and straight roads and contain less layout with each other. As the velocity of the vehicle was about 10 m/s and the maximum range of LiDAR detection was about 100 m, a 10-s interval was set to get a sequence of LiDAR frames. Accuracy, precision, recall and false positive rate (FPR) were applied to evaluate the performance of the training neural network. The performance of the pole classification neural network is shown in Table 3. The average time consumption of this module is about 0.03 s—much less than the time interval of LiDAR data accumulation as mentioned in Section 2.2. It is also worth mentioning that the training data and test data were collected in July 2019, while the data tested for the performance of the navigation module was collected in April 2019 in the next section. Therefore, the generalization and effectiveness of the pole extraction module will be validated.

Table 3. Performance of the artificial neural network (ANN) in the pole/nonpole classification task. FPR: false positive rate.

Accuracy	91.3%
Precision	88.7%
Recall	93.2%
FPR	10.3%

In the integrated navigation module, 16 mimic GNSS outages in two series of data were set to assess the performance. The shortest mileage of the testing platform during these outages was about 600 m, and the longest was about 1193 m in the plane, as shown in Figure 6.

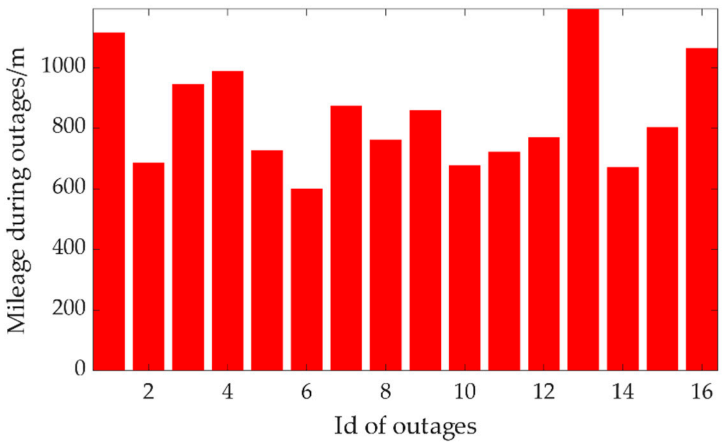
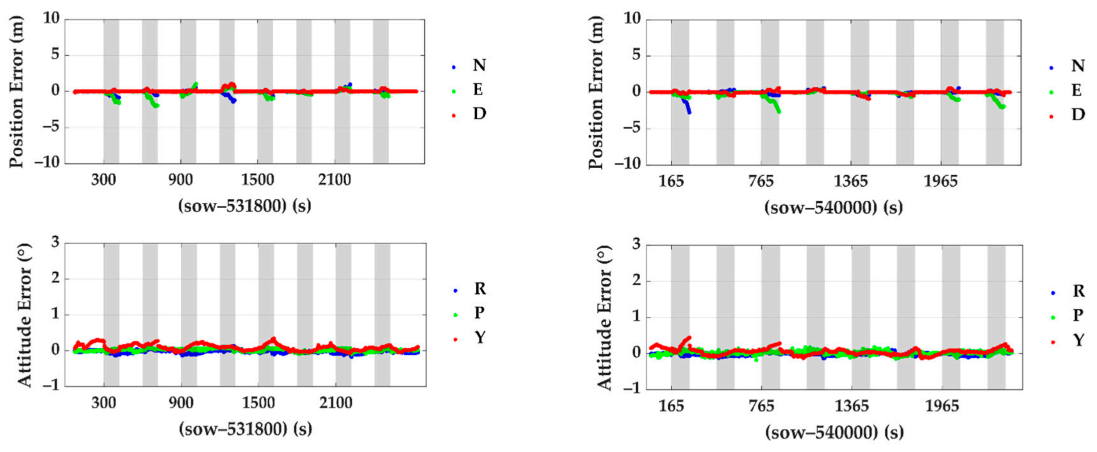


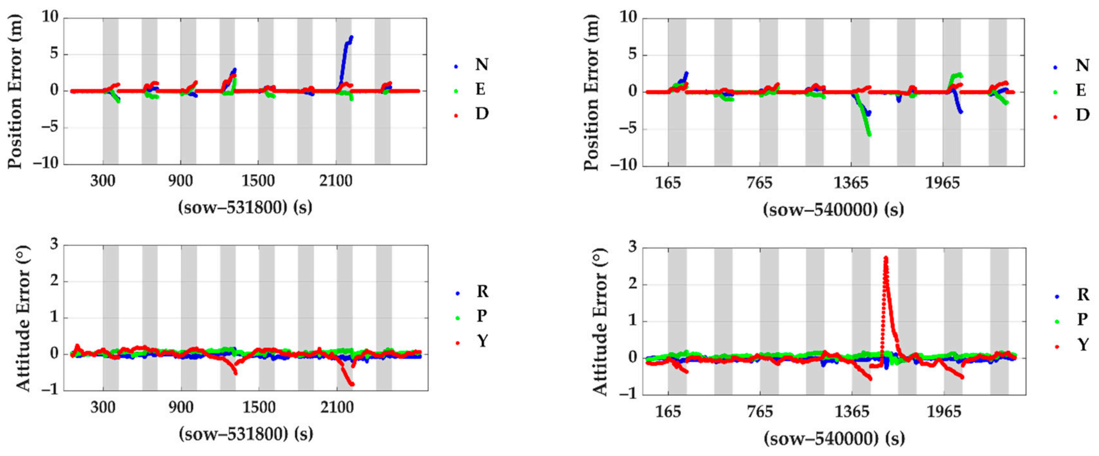
Figure 6. Mileage of the vehicle during mimic GNSS outages.

The benchmark method in Chang et al. [47] (Feature Point Grid-Based SLAM, called FPG SLAM in the rest of this paper) are analyzed here to make a comparison with the proposed method. Position errors and attitude errors during outages of the proposed method and benchmark are statistically analyzed. The details of the pose estimation errors and localization errors in every outage period are plotted in Figures 7 and 8. The grey spans in those figures mark the mimic GNSS outage periods.



(**a**) (**b**)

Figure 7. Position and attitude errors of the proposed integrated navigation system during GNSS outages: (**a**) errors of test #1 and (**b**) errors of test #2. The titles N, E and D represent the north, east and down directions, respectively, and R, P and Y mean roll, pitch and yaw.



(**a**) (**b**)

Figure 8. Position and attitude errors of the Feature Point Grid-Based Simultaneous Localization and Mapping (FPG SLAM) during the GNSS outages: (**a**) errors of test #1 and (**b**) errors of test #2.

The statistical results are also demonstrated in Table 4. The top rows are the statistic results of the benchmarked system. The bottom two rows are the results of the proposed method. The titles N, E and D represent the north, east and down directions, respectively, and R, P and Y mean roll, pitch and yaw. To provide an intuitive impression, the position errors of the traditional integrated navigation system composed of GNSS/IMU during the same outages are also calculated. The RMS (root mean square) errors in the north, east and down directions are 33.22 m, 23.47 m and 3.65 m, respectively, which are several times larger than the drift of the proposed method and FPG SLAM solution.

Table 4. Statistical results of the navigation errors obtained from different solutions during mimic GNSS outages. N, E and D indicate the position errors in the north, east and down, while R, P and Y represent the roll, pitch and yaw errors. FPG SLAM: Feature Point Grid-Based Simultaneous Localization and Mapping.

Solution		**Position Error (m)**			**Relative Plane Error**	**Attitude Error (°)**		
		N	**E**	**D**		**R**	**P**	**Y**
FPG SLAM	RMS	2.37	1.79	1.08	0.26%	0.12	0.14	0.34
	MAX	7.37	5.69	2.14		0.18	0.19	0.82
Proposed Method	RMS	0.91	1.22	0.53	0.16%	0.10	0.09	0.22
	MAX	2.69	2.64	1.10		0.14	0.12	0.45

Compared to FPG SLAM, both the maximum error and root mean square error of the position and attitude of the proposed method are smaller. In detail, the position error RMS was reduced by 61.6% and 31.8% in the north and east. The yaw angle error RMS was reduced by 35.3%. In general, it is more robust in some cases than the FPG SLAM, especially in congested environments where the target vehicle (or the LiDAR) is surrounded by other cars and pole-like objects are rich in the environment. Take the fourth and seventh outages in test #1, as shown in Figure 8a. The following Figure 9 shows the point cloud accumulated at around the 533,020th second of the week (contained in outage 4 in test #1). The white points are the point cloud, and the poles extracted by the pole extraction module are the red ones. The green boxes show the positions of the surrounding vehicles. The light green arrows show the directions of the roads. The orange dotted lines show the borders of the accessible areas of vehicles. In FPG SLAM, the feature points are extracted from the surrounding cars' reflections, while they deteriorate the estimation of the heading angle and position since the relative motion.

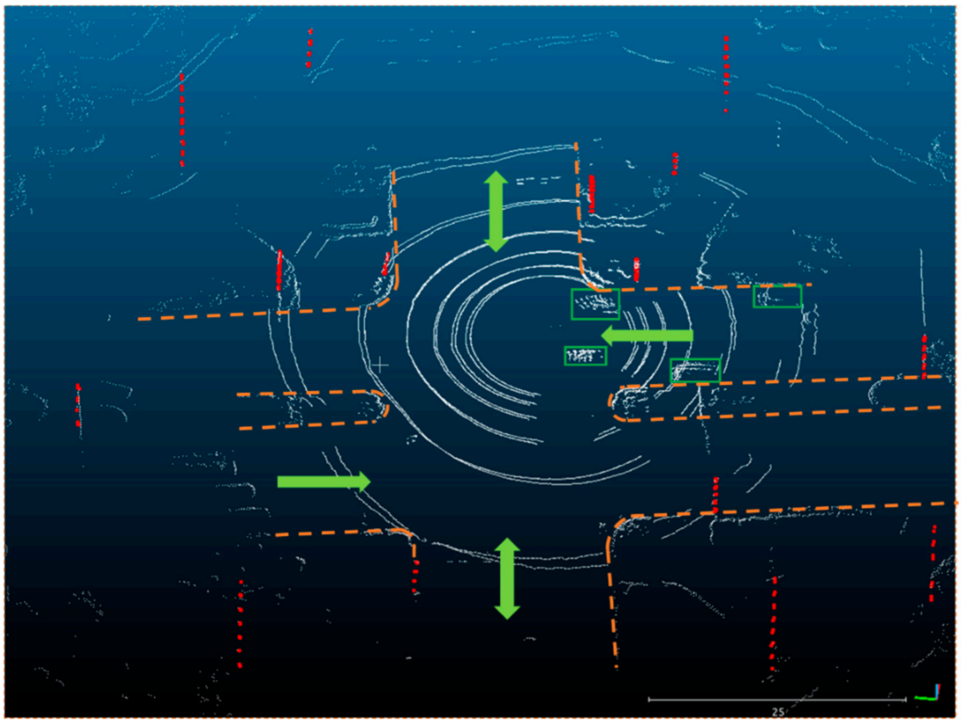


Figure 9. Point cloud at the 533,020th second of the week (in the 4th outage in Figure 8a). The red points are poles extracted by the algorithm mentioned above in Section 2.2; points in green boxes are laser reflections of the surrounding cars, and the point clouds are shown in white. The light green arrows are the directions of roads, and the orange dotted lines are the borders of accessible area.

At about the 533,920th second of the week (which was contained in outage 7 in test #1), the testing vehicle was surrounded by several cars at a crossroad, as shown in Figure 10. The feature points obtained from the laser reflections of the cars damaged the solutions of the position and attitude optimizations.

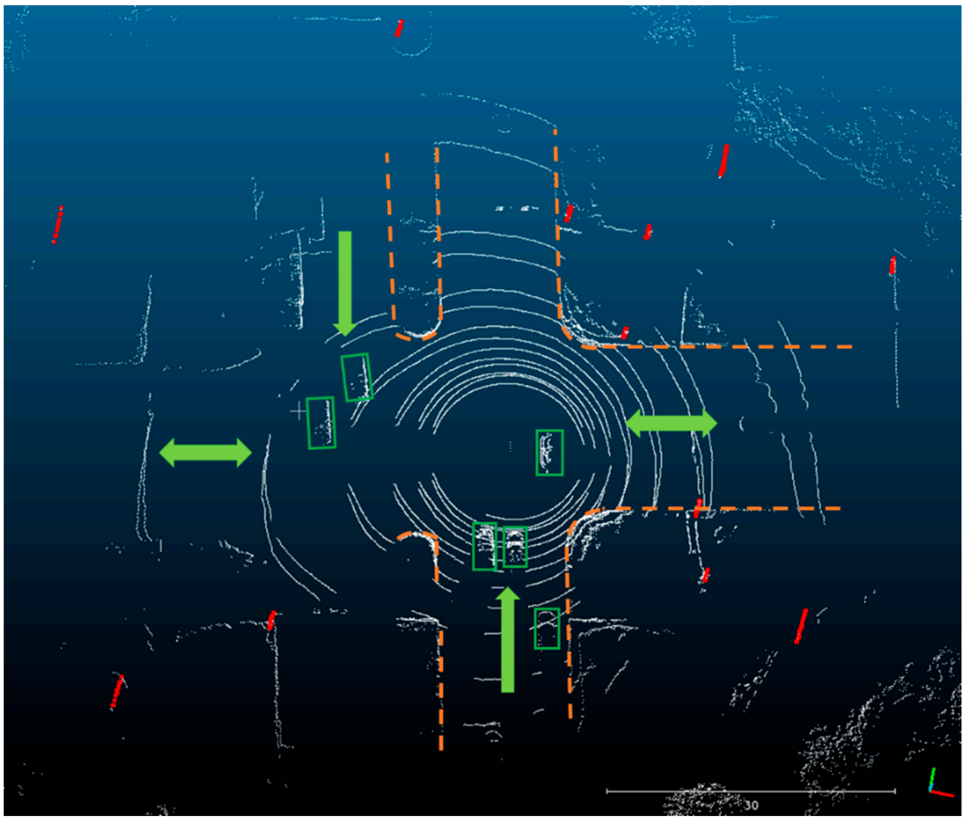
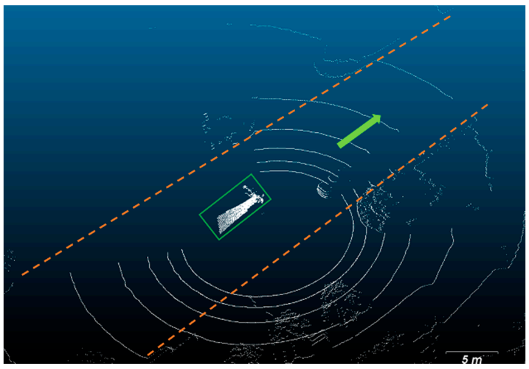


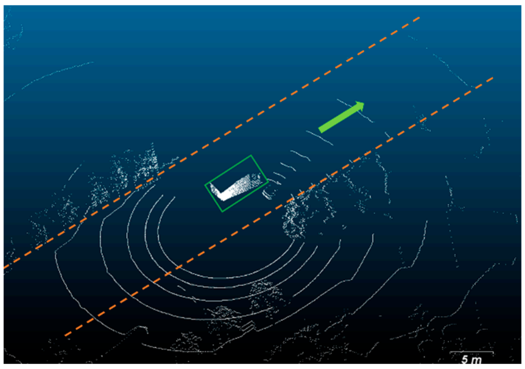
Figure 10. Point cloud at the 533,920th second of the week (in the 7th outage in Figure 8a). Cars in this environment were distributed in different distances and directions, and there existed relative motions between the neighbor cars and the testing vehicle. The color scheme is the same as in Figure 9.

In the FPG SLAM solution, there are a series of salient yaw errors in test #2, as shown in Figure 8b. Tracing back to the real case, the relative position of the testing vehicle and the van changed over time in four stages, and four figures are taken as examples here in Figure 11:

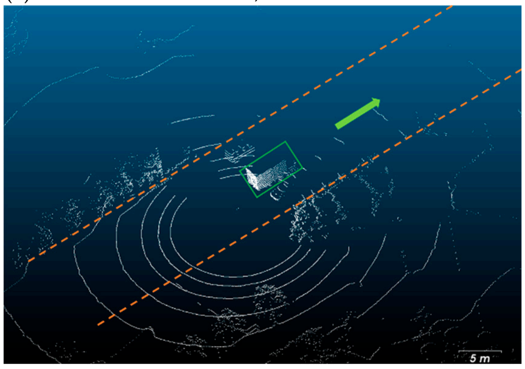
1. The testing vehicle stopped behind a car, and a van got closer and closer to it from the left-behind (shown in Figure 11a).
2. The van passed the testing vehicle, and the tail of the carriage appeared in sight (shown in Figure 11b).
3. The van slowed down and then stopped at the left-front of the testing vehicle for a while (shown in Figure 11c).
4. Both of them restarted moving, and the van disappeared gradually (shown in Figure 11d).



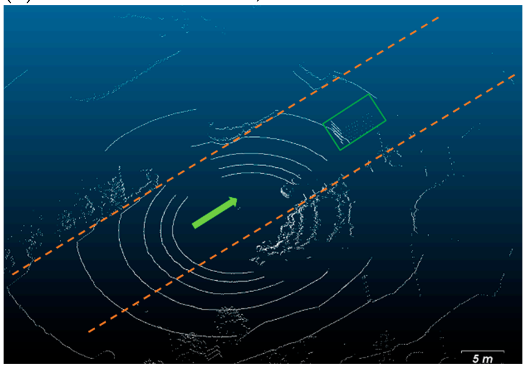
(**a**) Point cloud at the 541,556th second of the week.



(**b**) Point cloud at the 541,559th second of the week.



(**c**) Point cloud at the 541,575th second of the week.



(**d**) Point cloud at the 541,580th second of the week.

Figure 11. Point cloud collected when an abnormal heading angle error occurred. The laser reflections from the carriage of a van are in the green box. The light green arrow in each figure shows the driving direction of the vehicles, and the orange dotted lines show the borders of the road. The van was at the front-left of the experimental vehicle. (**a**–**d**) show the point cloud at different times, and they disappeared in sequence.

The relative position of the testing vehicle and the van is shown in Figure 12.

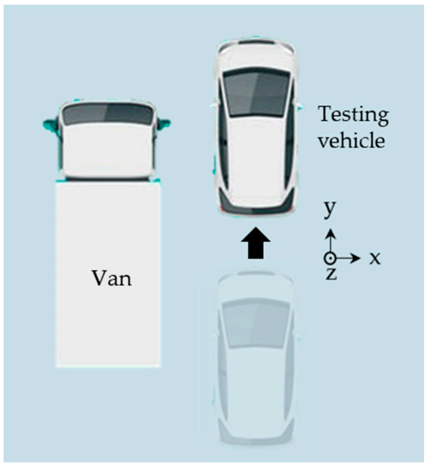


Figure 12. Relative position and motion of the testing vehicle and the neighbor van in a bird eye's view. The coordinates in the figure show the direction of each axis of the LiDAR. The testing vehicle moved nearly along the y-axis, as the misaligned angle between the vehicle and LiDAR was very small.

In this paper, we defined the angle error as

$$\Delta\theta_{yaw} = \theta_{yaw} - \hat{\theta}_{yaw}, \tag{19}$$

where $\Delta\theta_{yaw}$ is the error of the yaw angle, θ_{yaw} is the ground truth, which increases clockwise and $\hat{\theta}_{yaw}$ is the estimation of the yaw angle. A large amount of plane feature points were extracted from the laser reflections of the carriage tail of the van. In stage 2, the distance between the van and the vehicle was too close to make a drifted plane position estimation, while, to match the surface points to their corresponding previous LiDAR frames, the testing vehicle was estimated to turn left, resulting in a smaller yaw angle, as shown in Figure 13.

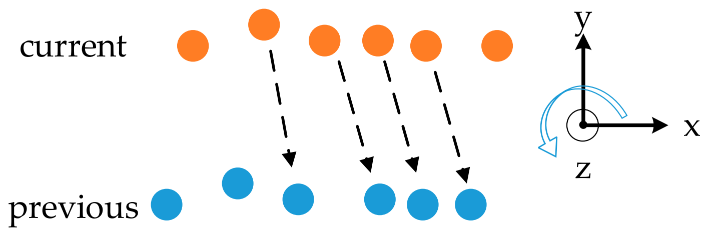


Figure 13. Plane feature points matched to their corresponding ones extracted from the previous LiDAR frame. The orange points are the plane feature points in the current frame, and the blue dots are the plane feature points in the previous frame or feature map. The black dotted line shows some of the correspondence of the feature points. The coordinates on the right show the direction of each axis of the l-frame.

Thus, positive yaw angle errors appeared in this period of time. In these cases, the LiDAR was surrounded by vehicles with smooth surfaces, such as vans, buses and trucks, along with relative motions among them so that plane feature points were extracted from those moving objects, resulting in bad constraints being added to the optimization problem. In a congested urban area, it is more likely that vans and trucks appear within sight of the LiDAR, causing a wrong estimation of orientation or position of the vehicles.

5. Conclusions

To handle the navigation problem in a GNSS-denied environment and take advantage of an urban infrastructure, this paper explored a pole extraction method using sparse LiDAR data and proposed a corresponding robust integrated navigation system composed of GNSS/IMU/LiDAR in an urban area. Experiments were carried out to evaluate the effectiveness of each module in the proposed solution, including the pole extraction module and integrated navigation module. In the validation of the pole extraction module, more than 800 samples were labeled manually. There were 80% of them were used for training a neural network, and 20% of them were used as the test set. The accuracy, precision, recall and FPR of the test set were 91.3%, 88.7%, 93.2% and 10.3%, respectively. The validity of the pole extraction module was verified in the navigation module, which took the coordinates of the poles as the feature points to do scan matching and then posed an optimization. In the integrated navigation module, a benchmark solution—FPG SLAM, as mentioned above, was demonstrated to make a comparison with the proposed method. The pole-based GNSS/IMU/LiDAR method outperformed the FPG SLAM integrated navigation system in terms of the position estimation and attitude estimation during the mimic GNSS outages. Compared with FPG SLAM, the position error of the proposed method was reduced by 61.6% and 31.8% in the north and east, and the yaw angle error RMS was reduced by 35.3%, especially in congested areas such as the crossroad, and the pole-based LiDAR SLAM showed its robustness and reliability in these cases. For future works, more features extracted from an urban infrastructure will be used to complement the insufficiency of the pole-like objects in some areas and enhance the performance of LiDAR-based multi-sensor SLAM.

Author Contributions: T.L. collected the data of the experiments, conceived and designed the experiments, performed the experiments and wrote the paper; X.N. and L.C. guided the design of the system algorithm, helped make the results analysis and approved the paper and J.L. provided guidance for the research direction

and offered a platform for the experiments. All authors have read and agreed to the published version of the manuscript.

Funding: Joint Fund of the Ministry of Education (6141A02011907) and The National Key Research and Development Program of China (2018YFB1305001).

Acknowledgments: This paper would not have been possible without the collaboration of our workmates. We thank Hailiang Tang for his kindness in providing the InsProbe device for data collection. The engineers in the Wuhan MAP Space Time Navigation Technology Corporation helped with hardware debugging on the M40 device. Juxiang Jiang and Yan Wang contributed their intelligence in discussions. We also thank Suhan Huang, Lexiang Zhang and Junwei Du for their help in the preparation of the experiments. Xin Feng and Jing Fan should also be appreciated for their work in proofreading.

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

References

1. Li, R.; Zheng, S.; Wang, E.; Chen, J.; Feng, S.; Wang, D.; Dai, L. Advances in BeiDou Navigation Satellite System (BDS) and satellite navigation augmentation technologies. *Satell. Navig.* **2020**, *1*, 12. [CrossRef]
2. Yang, Y.; Mao, Y.; Sun, B. Basic performance and future developments of BeiDou global navigation satellite system. *Satell. Navig.* **2020**, *1*, 1–8. [CrossRef]
3. Xu, H.; Angrisano, A.; Gaglione, S.; Hsu, L.-T. Machine learning based LOS/NLOS classifier and robust estimator for GNSS shadow matching. *Satell. Navig.* **2020**, *1*, 1–12. [CrossRef]
4. Shin, E.-H.; El-Sheimy, N. Accuracy improvement of low cost INS/GPS for land applications. In Proceedings of the 2002 National Technical Meeting of the Institute of Navigation, San Diego, CA, USA, 28–30 January 2002; pp. 146–157.
5. Li, W.; Cui, X.; Lu, M. A robust graph optimization realization of tightly coupled GNSS/INS integrated navigation system for urban vehicles. *Tsinghua Sci. Technol.* **2018**, *23*, 724–732. [CrossRef]
6. Wen, W.; Pfeifer, T.; Bai, X.; Hsu, L.-T. It is time for Factor Graph Optimization for GNSS/INS Integration: Comparison between FGO and EKF. *arXiv* **2019**, arXiv:2004.10572. Available online: https://arxiv.org/abs/2004.10572 (accessed on 30 November 2020).
7. Wen, W.; Bai, X.; Kan, Y.C.; Hsu, L.-T. Tightly coupled GNSS/INS integration via factor graph and aided by fish-eye camera. *IEEE Trans. Veh. Technol.* **2019**, *68*, 10651–10662. [CrossRef]
8. Shin, E.-H. Estimation Techniques for Low-Cost Inertial Navigation. Ph.D. Thesis, The University of Calgary, Calgary, AB, Canada, May 2005.
9. El-Sheimy, N.; Youssef, A. Inertial sensors technologies for navigation applications: State of the art and future trends. *Satell. Navig.* **2020**, *1*, 1–21. [CrossRef]
10. Fu, Q.; Yu, H.; Wang, X.; Yang, Z.; Zhang, H.; Mian, A. FastORB-SLAM: A Fast ORB-SLAM Method with Coarse-to-Fine Descriptor Independent Keypoint Matching. *arXiv* **2020**, arXiv:2008.09870. Available online: https://arxiv.org/abs/2008.09870 (accessed on 30 November 2020).
11. Jiang, J.; Niu, X.; Guo, R.; Liu, J. A hybrid sliding window optimizer for tightly-coupled vision-aided inertial navigation system. *Sensors* **2019**, *19*, 3418. [CrossRef]
12. Jiang, J.; Niu, X.; Liu, J. Improved IMU Preintegration with Gravity Change and Earth Rotation for Optimization-Based GNSS/VINS. *Remote Sens.* **2020**, *12*, 3048. [CrossRef]
13. Campos, C.; Elvira, R.; Rodríguez, J.J.G.; Montiel, J.M.; Tardós, J.D. ORB-SLAM3: An accurate open-source library for visual, visual-inertial and multi-map SLAM. *arXiv* **2020**, arXiv:2007.11898. Available online: https://arxiv.org/abs/2007.11898 (accessed on 30 November 2020).
14. Mur-Artal, R.; Montiel, J.M.M.; Tardos, J.D. ORB-SLAM: A versatile and accurate monocular SLAM system. *IEEE Trans. Robot.* **2015**, *31*, 1147–1163. [CrossRef]
15. 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]
16. 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 (ICRA), Brisbane, Australia, 21–25 May 2018; pp. 2480–2485.
17. 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]

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.* **2017**, *9*, 3. [CrossRef]
19. Chiang, K.-W.; Tsai, G.-J.; Li, Y.-H.; Li, Y.; El-Sheimy, N. Navigation Engine Design for Automated Driving Using INS/GNSS/3D LiDAR-SLAM and Integrity Assessment. *Remote Sens.* **2020**, *12*, 1564. [CrossRef]
20. Hess, W.; Kohler, D.; Rapp, H.; Andor, D. Real-time loop closure in 2D LIDAR SLAM. In Proceedings of the 2016 IEEE International Conference on Robotics and Automation (ICRA), Stockholm, Sweden, 16–21 May 2016; pp. 1271–1278.
21. Meng, X.; Wang, H.; Liu, B. A robust vehicle localization approach based on gnss/imu/dmi/lidar sensor fusion for autonomous vehicles. *Sensors* **2017**, *17*, 2140. [CrossRef]
22. Dubé, R.; Cramariuc, A.; Dugas, D.; Nieto, J.; Siegwart, R.; Cadena, C. SegMap: 3d segment mapping using data-driven descriptors. *arXiv* **2018**, arXiv:1804.09557. Available online: https://arxiv.org/abs/1804.09557 (accessed on 30 November 2020).
23. Dubé, R.; Dugas, D.; Stumm, E.; Nieto, J.; Siegwart, R.; Cadena, C. Segmatch: Segment based place recognition in 3d point clouds. In Proceedings of the 2017 IEEE International Conference on Robotics and Automation (ICRA), Singapore, 29 May–3 June 2017; pp. 5266–5272.
24. Ye, H.; Chen, Y.; Liu, M. Tightly coupled 3D LIDAR inertial odometry and mapping. In Proceedings of the 2019 International Conference on Robotics and Automation (ICRA), Singapore, 29 May–3 June 2017; pp. 3144–3150.
25. Zhao, S.; Farrell, J.A. 2D LIDAR aided INS for vehicle positioning in urban environments. In Proceedings of the 2013 IEEE International Conference on Control Applications (CCA), Hyderabad, India, 28–30 August 2013; pp. 376–381.
26. Im, J.-H.; Im, S.-H.; Jee, G.-I. Vertical corner feature based precise vehicle localization using 3D LIDAR in urban area. *Sensors* **2016**, *16*, 1268. [CrossRef]
27. Schaefer, A.; Büscher, D.; Vertens, J.; Luft, L.; Burgard, W. Long-term urban vehicle localization using pole landmarks extracted from 3-D lidar scans. In Proceedings of the 2019 European Conference on Mobile Robots (ECMR), Prague, Czech Republic, 4–6 September 2019; pp. 1–7.
28. Weng, L.; Yang, M.; Guo, L.; Wang, B.; Wang, C. Pole-based real-time localization for autonomous driving in congested urban scenarios. In Proceedings of the 2018 IEEE International Conference on Real-time Computing and Robotics (RCAR), Kandima, Maldives, 1–5 August 2018; pp. 96–101.
29. Cabo, C.; Ordoñez, C.; García-Cortés, S.; Martínez, J. An algorithm for automatic detection of pole-like street furniture objects from Mobile Laser Scanner point clouds. *ISPRS J. Photogramm. Remote Sens.* **2014**, *87*, 47–56. [CrossRef]
30. Rodríguez-Cuenca, B.; García-Cortés, S.; Ordóñez, C.; Alonso, M.C. Automatic detection and classification of pole-like objects in urban point cloud data using an anomaly detection algorithm. *Remote Sens.* **2015**, *7*, 12680–12703. [CrossRef]
31. Yu, Y.; Li, J.; Guan, H.; Wang, C.; Yu, J. Semiautomated extraction of street light poles from mobile LiDAR point-clouds. *IEEE Trans. Geosci. Remote Sens.* **2014**, *53*, 1374–1386. [CrossRef]
32. Zheng, H.; Wang, R.; Xu, S. Recognizing street lighting poles from mobile LiDAR data. *IEEE Trans. Geosci. Remote Sens.* **2016**, *55*, 407–420. [CrossRef]
33. Wu, F.; Wen, C.; Guo, Y.; Wang, J.; Yu, Y.; Wang, C.; Li, J. Rapid localization and extraction of street light poles in mobile LiDAR point clouds: A supervoxel-based approach. *IEEE Trans. Intell. Transp. Syst.* **2016**, *18*, 292–305. [CrossRef]
34. Qi, C.R.; Su, H.; Mo, K.; Guibas, L.J. Pointnet: Deep learning on point sets for 3D classification and segmentation. In Proceedings of the IEEE conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 652–660.
35. Qi, C.R.; Yi, L.; Su, H.; Guibas, L.J. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In Proceedings of the Advances in Neural Information Processing Systems (NIPS), Long Beach, CA, USA, 4–9 December 2017; pp. 5099–5108.
36. Long, J.; Shelhamer, E.; Darrell, T. Fully convolutional networks for semantic segmentation. In Proceedings of the IEEE conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 8–10 June 2015; pp. 3431–3440.

37. Wu, B.; Wan, A.; Yue, X.; Keutzer, K. Squeezeseg: Convolutional neural nets with recurrent crf for real-time road-object segmentation from 3D lidar point cloud. In Proceedings of the 2018 IEEE International Conference on Robotics and Automation (ICRA), Brisbane, Australia, 21–25 May 2018; pp. 1887–1893.
38. Wu, B.; Zhou, X.; Zhao, S.; Yue, X.; Keutzer, K. Squeezesegv2: Improved model structure and unsupervised domain adaptation for road-object segmentation from a lidar point cloud. In Proceedings of the 2019 International Conference on Robotics and Automation (ICRA), Montreal, QC, Canada, 20–24 May 2019; pp. 4376–4382.
39. Zhou, Y.; Tuzel, O. Voxelnet: End-to-end learning for point cloud based 3d object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition(CVPR), Salt Lake City, UT, USA, 18–22 June 2018; pp. 4490–4499.
40. Teo, T.-A.; Chiu, C.-M. Pole-like road object detection from mobile lidar system using a coarse-to-fine approach. *IEEE J. Sel. Top. Appl. Earth Obs. Remote. Sens.* **2015**, *8*, 4805–4818. [CrossRef]
41. Zheng, H.; Tan, F.; Wang, R. Pole-like object extraction from mobile LIDAR data. In Proceedings of the International Archives of the Photogrammetry, Remote Sensing & Spatial Information Sciences (I ISPRS), Prague, Czech Republic, 12–19 July 2016; pp. 729–734.
42. Li, Y.; Wang, W.; Li, X.; Xie, L.; Wang, Y.; Guo, R.; Xiu, W.; Tang, S. Pole-Like Street Furniture Segmentation and Classification in Mobile LiDAR Data by Integrating Multiple Shape-Descriptor Constraints. *Remote Sens.* **2019**, *11*, 2920. [CrossRef]
43. Song, W.; Zou, S.; Tian, Y.; Fong, S.; Cho, K. Classifying 3D objects in LiDAR point clouds with a back-propagation neural network. *Hum. Cent. Comput. Inf. Sci.* **2018**, *8*, 1–12. [CrossRef]
44. Shan, T.; Englot, B. Lego-loam: Lightweight and ground-optimized lidar odometry and mapping on variable terrain. In Proceedings of the 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Madrid, Spain, 1–5 October 2018; pp. 4758–4765.
45. Liu, X.; Zhang, L.; Qin, S.; Tian, D.; Ouyang, S.; Chen, C. Optimized LOAM Using Ground Plane Constraints and SegMatch-Based Loop Detection. *Sensors* **2019**, *19*, 5419. [CrossRef]
46. Himmelsbach, M.; Hundelshausen, F.V.; Wuensche, H.-J. Fast segmentation of 3D point clouds for ground vehicles. In Proceedings of the 2010 IEEE Intelligent Vehicles Symposium, San Diego, CA, USA, 21–24 June 2010; pp. 560–565.
47. Chang, L.; Niu, X.; Liu, T. GNSS/IMU/ODO/LiDAR-SLAM Integrated Navigation System Using IMU/ODO Pre-Integration. *Sensors* **2020**, *20*, 4702. [CrossRef]
48. Chang, L.; Niu, X.; Liu, T.; Tang, J.; Qian, C. GNSS/INS/LiDAR-SLAM Integrated Navigation System Based on Graph Optimization. *Remote Sens.* **2019**, *11*, 1009. [CrossRef]