

Article A Method of Setting the LiDAR Field of View in NDT Relocation Based on ROI

Jian Gu^{1,2,†}, Yubin Lan^{1,2,3,†}, Fanxia Kong^{1,2}, Lei Liu^{1,2}, Haozheng Sun^{1,2}, Jie Liu^{1,2} and Lili Yi^{1,2,*}

- School of Agricultural Engineering and Food Science, Shandong University of Technology, Zibo 255000, China
- ² Shandong University of Technology Sub-Center of National Center for International Collaboration Research on Precision Agricultural Aviation Pesticide Spraying Technology, National Sub-Center for International Collaboration Research on Precision Agricultural Aviation Pesticide Spraying Technology, Zibo 255000, China
- ³ National Center for International Collaboration Research on Precision Agricultural Aviation Pesticides Spraying Technology, College of Engineering, Shandong University of Technology, Guangzhou 510642, China
- * Correspondence: yili0001@sdut.edu.cn
- + These authors contributed equally to this work.

Abstract: LiDAR placement and field of view selection play a role in detecting the relative position and pose of vehicles in relocation maps based on high-precision map automatic navigation. When the LiDAR field of view is obscured or the LiDAR position is misplaced, this can easily lead to loss of repositioning or low repositioning accuracy. In this paper, a method of LiDAR layout and field of view selection based on high-precision map normal distribution transformation (NDT) relocation is proposed to solve the problem of large NDT relocation error and position loss when the occlusion field of view is too large. To simulate the real placement environment and the LiDAR obstructed by obstacles, the ROI algorithm is used to cut LiDAR point clouds and to obtain LiDAR point cloud data of different sizes. The cut point cloud data is first downsampled and then relocated. The downsampling points for NDT relocation are recorded as valid matching points. The direction and angle settings of the LiDAR point cloud data are optimized using RMSE values and valid matching points. The results show that in the urban scene with complex road conditions, there are more front and rear matching points than left and right matching points within the unit angle. The more matching points of the NDT relocation algorithm there are, the higher the relocation accuracy. Increasing the front and rear LiDAR field of view prevents the loss of repositioning. The relocation accuracy can be improved by increasing the left and right LiDAR field of view.

Keywords: normal distribution transformation; high-precision maps; automatic navigation

1. Introduction

Over the past two decades, relocation has played a crucial role in automotive driving applications. Relocation is used to obtain the vehicle's global position and pose based on the built map. Various relocation methods have been successful in resolving such problems using global navigation satellite systems (GNSS) [1–3], the inertial measurement unit (IMU) [4,5], cameras, LiDAR, and other sensing sensors.

The integration of GNSS and IMU is a common solution to these problems [6–8]. However, GNSS is prone to signal interference or loss, in urban buildings, for example. The current solution is to use mileage information to compensate for GNSS measurements. However, performance in auto-driving related applications is still generally insufficient.

Cameras are one of the most attractive relocation sensors because of their inherent high information content, low cost, and small size [9–11]. Visual relocation uses the large amount of information provided by the camera to estimate the robot position. Because accurate and reliable positioning is required where the GPS signal is weak, a new visual measurement framework for land vehicle positioning was proposed by Zhiyong Zheng [12].



Citation: Gu, J.; Lan, Y.; Kong, F.; Liu, L.; Sun, H.; Liu, J.; Yi, L. A Method of Setting the LiDAR Field of View in NDT Relocation Based on ROI. *Sensors* **2023**, *23*, 843. https:// doi.org/10.3390/s23020843

Academic Editor: Aboelmagd Noureldin

Received: 3 November 2022 Revised: 2 December 2022 Accepted: 10 January 2023 Published: 11 January 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https:// creativecommons.org/licenses/by/ 4.0/). However, for outdoor large-scale relocation, ensuring its robust operation is still very challenging especially in changing environments and adverse weather conditions. Kaixin Yang presents a coordinated positioning strategy that is composed of semantic information and probabilistic data association, which improves the accuracy of SLAM (Simultaneous Localization and Mapping) in dynamic traffic settings [13]. However, many walls and windows in a city are made of glass or vinyl and are easily exposed to light during the day.

An increasingly popular solution to the city problem is to stop using GNSS and camera measurements altogether and rely on LiDAR, which measures 3D scans of the environment. Compared with cameras and GNSS, LiDAR has better penetrability and anti-interference characteristics. LiDAR detects objects and surfaces using distance measurements, which are computed from the time-of-flight of the reflected light pulses [14–16]. Furthermore, the amount of data acquired by a LiDAR system is less than in the case of a camera. Hence the LiDAR sensor is used frequently for positioning and object detection. However, most LiDAR-based localization solutions with prior point-cloud maps assume that the road scenes are relatively constant, therefore new constructions, road-side vegetation, partial occlusions by changing objects may severely compromise its robustness. Therefore, an interesting but open question is whether LiDAR can be used for robust relocalization in large-scale changing environments.

In large-scale and dynamic urban environments, the normal distribution transformation (NDT) algorithm based on high-precision maps is one of the main repositioning algorithms widely used in automotive driving applications [17,18]. In comparison to the SLAM algorithm, the NDT algorithm is found to be more effective for large point clouds. The NDT registration algorithm is time-consuming and stable, has little correlation with the initial value, and can be corrected well when the initial error is large. The open source Autoware algorithm at Nagoya University mainly uses the NDT algorithm. The Autoware algorithm uses the NDT matching registration algorithm to obtain its own position and position information in the LiDAR point cloud map. However, the NDT algorithm is not robust regarding significant geometric changes to the environment or overly unexpected or dynamic objects [19,20]. These shortcomings seriously affect the performance of scan matching based on high-precision maps. The NDT relocation algorithm can handle some environmental changes, such as the accidental obscuring of objects (LiDAR sensors partially obscured by leaves or other vehicles), which is unavoidable in cities. When there is a difference between the environment and the map, the positioning accuracy of nondestructive testing decreases. In practical applications multi-LiDAR is used for positioning. The position of the LiDAR directly determines the LiDAR field of view. Different views of the LiDAR field of view have a direct impact on the positioning accuracy of vehicles. Many studies have not specifically assessed the relationship between point cloud occlusion at different locations and noninvasive detection performance based on high-precision maps.

2. Algorithm Principle

2.1. Loam

LOAM (lidar odometry and mapping in real time) is a high-precision and real-time positioning and mapping algorithm based on 3D LiDAR proposed by Ji Zhang et al. [21,22]. The core of the LOAM framework is in two parts, a high frequency odometer and low frequency mapping, as shown in Figure 1.



Figure 1. System structure diagram of the LOAM algorithm.

The odometer performs scan-scan matching through a high frequency and low number of point clouds, estimates the motion relationship between two frames, and outputs the results to the mapping algorithm; mapping matches and aligns the undistorted point cloud to the map at a frequency of 1 Hz, using the scan-map matching method [23,24]. Finally, the attitude transformation created by the two algorithms is integrated to obtain the transformation output of LiDAR, with an attitude to the map of about 10 Hz.

We start with the extraction of feature points from the LiDAR cloud, p_k . We select feature points that are on sharp edges and planar surface patches. Let *i* be a point in p_k , $i \in p_k$ and let *S* be the set of consecutive points of *i* returned by the LiDAR scanner in the same scan. This defines a term to evaluate the smoothness of the local surface as

$$c = \frac{1}{|S| \cdot \left\| X_{(K,i)}^{L} \right\|} \left\| \sum_{j \in S, j \neq i} \left(X_{(K,i)}^{L} - X_{(K,j)}^{L} \right) \right\|.$$
(1)

In this paper, the curvature of a point is calculated according to the above formula. In practice, we only need to compare the curvature of a point so that we can find the curvature of the point in the square coordinate of the difference of five points around a point. In this way, we can find the curvature, *c*, of each point, and by comparing the curvatures, we can select the edge points with a larger curvature and the plane points with a smaller curvature. To prevent the feature points from clustering, each scanned point cloud is divided into four parts, from which two points with the largest curvature are selected as edge points and four points with the smallest curvature are selected as plane points.

When selecting points, we want to avoid selecting points around already selected points or points whose LiDAR lines are close to parallel planes, which are generally considered unreliable because they cannot be seen at any time. We also want to avoid possible obscuring points. The odometry algorithm estimates the motion of the LiDAR within a sweep. Let t_k be the starting time of a sweep, k. At the end of each sweep, the point cloud perceived during the sweep, p_k , is reprojected to the time stamp t_{k+1} . We denote the reprojected point cloud as \bar{p}_k . During the next sweep, k + 1, \bar{p}_k is used together with the newly received point cloud, p_{k+1} , to estimate the motion of the LiDAR. The next step is to find the corresponding relationship, i.e., to match the feature points of two point clouds; the corner point of p_{k+1} matches the corner line of \bar{p}_k , and the plane point of p_{k+1} matches the plane of \bar{p}_k . With the corresponding relationship between point-to-line and point-to-face, we can calculate the distance between point-to-line and point-to-face:

$$d_{E} = \frac{\left| \left(\tilde{X}_{(k+1,i)}^{L} - \bar{X}_{(k,j)}^{L} \right) \times \left(\tilde{X}_{(k+1,i)}^{L} - \bar{X}_{(k,l)}^{L} \right) \right|}{\left| \left(\bar{X}_{(k,j)}^{L} - \bar{X}_{(k,l)}^{L} \right) \right|}$$
(2)

Then we can find the distance from the plane point to the corresponding plane:

$$d_{H} \frac{\left| \begin{pmatrix} \tilde{X}_{(k+1,i)}^{L} - \bar{X}_{(k,j)}^{L} \\ \left| \left(\left(\bar{X}_{(k,j)}^{L} - \bar{X}_{(k,l)}^{L} \right) \times \left(\bar{X}_{(k,j)}^{L} - \bar{X}_{(k,m)}^{L} \right) \right) \right|}{\left| \left(\bar{X}_{(k,j)}^{L} - \bar{X}_{(k,l)}^{L} \right) \times \left(\bar{X}_{(k,j)}^{L} - \bar{X}_{(k,m)}^{L} \right) \right|}$$
(3)

The LiDAR motion is modelled with constant angular and linear velocities during a sweep. This allows us to linearly interpolate the pose transformation within a sweep for the points that are received at different times. If we let *t* be the current time stamp, and recall that t_{k+1} is the starting time of sweep k + 1, the linear interpolation formula is

$$T_{(k+1,i)}^{L} = \frac{t_i - t_{k+1}}{t - t_{k+1}} T_{k+1}^{L}.$$
(4)

In order to obtain the corresponding relationship between the points in this frame data and the points in the previous frame data, we use a rotation matrix *R* and a translation amount *T*.

$$X_{(k+1,i)}^{L} = R\tilde{X}_{(k+1,i)}^{L} + T_{(k+1,i)}^{L}(1:3)$$
(5)

Since the derivation of rotation matrix is very complex, the rotation matrix *R* is expanded as follows by the Rodrigues formula:

$$R = e^{\hat{\omega}\theta} = I + \hat{\omega}\sin\theta + \hat{\omega}^2(1 - \cos\theta).$$
(6)

This makes it easy to derive the rotation matrix.

Now we have the distance from point-to-line and point-to-face, we can obtain the error function for optimization:

$$f\left(T_{k+1}^L\right) = d. \tag{7}$$

Each line in f represents a characteristic point. The next requirement is to solve the Jacobian matrix. Finally, the LM method is used for optimization:

$$T_{k+1}^{L} \leftarrow T_{k+1}^{L} - \left(J^{T}J + \lambda diag\left(J^{T}J\right)\right)^{-1}J^{T}d$$

$$\tag{8}$$

Since the solution in the previous step is the result according to the local LiDAR observation coordinate system T^L , it solves the transformation between adjacent frames. However, in order to simultaneously locate and map, it is necessary to solve the transformation under the global coordinate system T^W . Therefore, when we obtain the attitude transformation information of several adjacent frames, we need to match it with the global map and add it to the global map.

Finally, the attitude information obtained from the LiDAR odometer solution and the information obtained from the map construction are transformed and integrated, through the use of rviz software, for example.

2.2. NDT Relocation Algorithm

In order to identify the location of the LiDAR in the offline map, we compared the point cloud from the LiDAR scan with the point cloud from the offline map. During the relocation process, the point cloud from the LiDAR scan may differ from the point cloud from the offline map, either because the LiDAR field of view is occluded, or because the vehicle uses only part of the LiDAR point cloud.

For relocation in maps with deviations, we use the NDT alignment algorithm, which does not compare the difference between two point clouds but transforms the reference point cloud map into a normal distribution of multidimensional variables [25–27]. If the transformation parameters enable a good match between the two sets of LiDAR data, then

the probability density of the transformed points in the reference system will be large. Therefore, an optimization method can be considered to find the transformation parameter that maximizes the sum of the probability densities, when the two sets of LiDAR point cloud data will match best.

The first step is to grid the 3D offline point cloud map, using a small cube to divide the entire space of scanned points, and for each grid, calculate its probability density function based on the points within the grid. This can be described as:

$$\vec{u} = \frac{1}{m} \sum_{q=1}^{m} \vec{y}_q \tag{9}$$

$$\Sigma = \frac{1}{m} \sum_{q=1}^{m} \left(\vec{y}_q - \vec{\mu} \right) \left(\vec{y}_q - \vec{\mu} \right)^T$$
(10)

where $\vec{\mu}$ is the mean of the normal distribution of the grids of the offline map, *m* indicates the number of points in the offline map grid, *q* means the *q*th point in the offline map grid, $\vec{y}_{q=1,...,m}$ for all scanned points in the offline map grid, and \sum denotes the covariance matrix of the offline map grid. The probability density function of a grid can be described as:

$$f(\vec{x}) = \frac{1}{(2\pi)^{\frac{3}{2}}\sqrt{|\Sigma|}} e^{-\frac{(\vec{x}-\vec{\mu})^T \Sigma^{-1}(\vec{x}-\vec{\mu})}{2}}.$$
(11)

The use of normal distribution to represent an otherwise discrete offline point cloud map has many benefits. This chunked, smooth representation is continuously derivable and the probability density function of each lattice can be thought of as an approximation to a local surface, which not only describes the location of the surface in space, but also contains information about the orientation and smoothness of the surface.

When using NDT alignment, the goal is to find the pose of the current LiDAR scan in such a way as to maximize the likelihood of the currently scanned points lying on the surface of the offline map. The parameter we then need to optimize is the transformation (rotation, translation, etc.) of the currently scanned LiDAR point cloud, which we describe using a transformation parameter \vec{h} . The current scan is a point cloud $X = {\vec{x}_1, \ldots, \vec{x}_n}$, given the set of scan points X and the transformation parameter \vec{h} , such that the spatial transformation function $T(\vec{h}, \vec{x}_q)$ denotes the use of the pose transformation \vec{h} to move the points \vec{x}_q , combined with the previous set of density-of-state functions (Probability Density Function for each grid), then the best transformation parameter \vec{h} should be the pose transformation that maximizes the likelihood function:

$$Likelihood: \Theta = \prod_{q=1}^{n} f\left(T\left(\vec{h}, \vec{x}_{q}\right)\right).$$
(12)

Then, maximizing the likelihood is also equivalent to minimizing the negative log-likelihood $-\log \Theta$;

$$-\log\Theta = -\sum_{q=1}^{n}\log\Big(f\Big(T\Big(\vec{h},\vec{x}_q\Big)\Big)\Big).$$
(13)

An optimization algorithm is then used to tune the transformation parameter h to minimize this negative log likelihood. YThe NDT algorithm uses Newton's method for parameter optimization. Here the probability density function $f(\vec{x})$ does not have to be normally distributed; any probability density function that reflects the structural information of the scanned surface and is robust to anomalous scan points will be sufficient.

2.3. Point Cloud Data Preprocessing

The positive direction of the x-axis is the front of the LiDAR, and the positive direction of the y-axis is the left side of the LiDAR, as shown in Figure 2. Different LiDAR point cloud areas are extracted through the ROI (area of interest), and LiDAR point cloud areas of different sizes are reserved at the front, back, left, and right to simulate the changes of the LiDAR field of view in different degrees. The ROI can be delineated for further processing. The LiDAR point clouds in each region can be described as:



Figure 2. Simulation of different fields of view of LiDAR.

$$f_{\rm f}(\alpha) = \begin{cases} [(a,b)|b-a\tan\beta < 0] \cap [(a,b)|b+a\tan\beta > 0], & \alpha \in [0^{\circ}, 180^{\circ}]\\ [(a,b)|b-a\tan\beta < 0] \cup [(a,b)|b+a\tan\beta > 0], & \alpha \in (180^{\circ}, 360^{\circ}] \end{cases} ,$$
(14)

$$f_b(\alpha) = \begin{cases} [(a,b)|b-a\tan\beta > 0] \cap [(a,b)|b+a\tan\beta < 0], & \alpha \in [0^\circ, 180^\circ] \\ [(a,b)|b-a\tan\beta > 0] \cup [(a,b)|b+a\tan\beta < 0], & \alpha \in (180^\circ, 360^\circ] \end{cases},$$
(15)

$$f_{l}(\alpha) = \begin{cases} [(a,b)|b-a\tan\beta > 0] \cap [(a,b)|b+a\tan\beta > 0], & \alpha \in [0^{\circ}, 180^{\circ}]\\ [(a,b)|b-a\tan\beta > 0] \cup [(a,b)|b+a\tan\beta > 0], & \alpha \in (180^{\circ}, 360^{\circ}] \end{cases}$$
(16)

$$f_r(\alpha) = \begin{cases} [(a,b)|b-a\tan\beta < 0] \cap [(a,b)|b+a\tan\beta < 0], & \alpha \in [0^\circ, 180^\circ] \\ [(a,b)|b-a\tan\beta < 0] \cup [(a,b)|b+a\tan\beta < 0], & \alpha \in (180^\circ, 360^\circ] \end{cases}$$
(17)

The front, back, left, and right of the LiDAR point cloud areas are represented by $f_f(\alpha)$, $f_b(\alpha)$, $f_l(\alpha)$, $f_r(\alpha)$. The point inside the LiDAR point cloud is represented by (a, b), α is the angle of view of the cut LiDAR point cloud, and β is from 0° to 90°, as shown in Figure 3.



Figure 3. Simulation of the different fields of view of LiDAR.

Voxel downsampling creates a 3D voxel grid (considering the voxel grid as a collection of spatial 3D cubes) from the input point cloud data [28–31]. Then within each voxel (3D cube), the other points in the voxel are approximated by the center of gravity of all points in the voxel, so that all points in the voxel are represented by one center-of-gravity point. Voxel downsampling creates a 3D voxel grid (considering the voxel grid as a collection of spatial 3D cubes) from the input point cloud data. Then within each voxel (3D cube), all points in the voxel are approximated using the voxel's center of gravity to reveal other points in the voxel so that all points in the voxel are represented by a single center of gravity point. As a result, the number of points is reduced, the matching speed is improved, the shape features of the point cloud remain basically unchanged, and the spatial structure information is preserved. The larger the voxel grid selection, the smaller the sampled point cloud, and the faster the processing speed; however, the original point cloud will be too blurred. A smaller voxel grid selection will have the opposite effect. At the same time, it is necessary to record the number of points of the different LiDAR point cloud data after desampling, as shown in Figure 4.



Figure 4. LiDAR point cloud data downsampling.

3. The KITTI Dataset Test

To test the effect of the different fields of view of LiDAR on the NDT relocalization algorithm, we used the KITTI dataset with a full length of 864.831 m and a duration of 117 s. The test platform was a Velodyne HDL-64E-equipped vehicle. All experiments were performed on this platform. The average speed of the vehicle was about 2.5 m/s. All the evaluation experiments were run on a computer with an AMD R7-4800H processor, 16 GB RAM, and a single NVIDIA GeForce GTX1650ti GPU. Velodyne HDL-64E is a 64-line digital LiDAR mounted directly above the mobile chassis, with a 360° horizontal field of view, 5–15 Hz rotational speed, a 26.8° vertical field of view (+2° to -24.8°), vertical angular resolution of 0.4°, horizontal angular resolution of 0.08°, a point cloud count up to 1.3 million points per second, a maximum range of 100 m, and a ranging accuracy of ±2 cm.

As shown in Figure 5, a high-precision map is constructed by using the loam algorithm for the KITTI dataset. The NDT matching points are the points where the original point cloud data has been ROI processed and downsampled, as shown in Figure 6. Due to the uneven density of the original point cloud after voxel downsampling, the number of NDT matching points in each direction is also irregular, so the positioning accuracy of the NDT matching points in each direction is different. The downsampling factor set by the code in this article is 3.0. The transformation epsilon is 0.05, the step size is 0.1, the resolution is 2.0, and the maximum number of iterations is 30. This paper uses root mean squared error (RMSE) to measure the positioning accuracy. The RMSE is the square of the ratio of the square of the deviation between the predicted value and the true value and the number of observations.



Figure 5. The high-precision maps of KITTI datasets.



Figure 6. Local magnification during NDT relocation in KITTI datasets.

As shown in Table 1, the RMSE obtained from the LiDAR point cloud reposition tracks at 90°, 180°, and 270° in different directions is compared. It can be seen that in the same direction, the larger the view angle of the LiDAR and the more effective the matching points of the NDT algorithm, the higher the positioning accuracy and the smaller the error. The LiDAR cannot be repositioned when the left and right LiDAR field of view angles are 90 degrees. When the front and back LiDAR field of view angle is 90°, it can be repositioned. When the angle of the LiDAR field of view in four directions is enlarged, the precision of the front and back sides is improved greatly, while the precision of the left and right sides is improved less.

In order to further explore the effect of the LiDAR-matching point cloud orientation and the number of NDT matching points on the positioning accuracy and weight in the point cloud, the influence of the positioning trajectory drift is extended in the middle of 90° to 270° on the left and right. Experiments are carried out at different angles at 30 degrees. As shown in Table 2, by comparing the number of NDT matching points on the left and right sides of the LiDAR, it can be seen that due to the large number of NDT-matching point clouds at the front and the back, and at 90°–270° on the left and right sides, at 1°, the proportion of the number of matching points before and after NDT plays a decisive role in the accuracy of the positioning trajectory. At the same time, it is also proved that the more NDT matching points there are, the higher the positioning accuracy and the smaller the degree of drift. In normal driving mode, since the point clouds at the front and back of the radar point cloud are relatively abundant, it is recommended to use the front and back point clouds more.

	Max (m)	Mean (m)	Min (m)	Rmse (m)	Num (m)	Ratio (%)
90_front	1.803023	0.820853	0.048090	0.953158	224,995	34.7
180_front	0.468620	0.213139	0.019969	0.224281	316,011	48.7
270_front	0.323754	0.115183	0.010731	0.126000	423,800	65.3
90_back	1.081605	0.646901	0.023327	0.690814	238,414	36.8
180_back	0.930584	0.452172	0.024949	0.500723	332,584	51.3
270_back	0.600951	0.250809	0.023253	0.276960	437,715	67.5
90_left	0	0	0	0	117,606	18.1
180_left	0.492303	0.244453	0.018364	0.265370	337,150	52.0
270_left	0.310869	0.132304	0.014551	0.141764	566,296	87.3
90_right	0	0	0	0	7658	0.12
180_right	1.422529	0.739286	0.022606	0.804200	311,443	48.0
270_right	1.029491	0.498957	0.009679	0.562762	545,708	84.1

Table 1. Absolute track error in four directions.

Table 2. Absolute trajectory error left and right.

	Max (m)	Mean (m)	Min (m)	Rmse (m)	Num (m)	Ratio (%)
90_left	0	0	0	0	117606	18.1
120_left	1.408121	0.533117	0.010324	0.576745	162,363	25.0
150_left	1.042893	0.504791	0.013068	0.549890	223,012	34.5
180_left	0.492303	0.244453	0.018364	0.265370	337,150	52.0
210_left	0.425172	0.164817	0.012010	0.177659	482,429	74.4
240_left	0.374192	0.143122	0.009739	0.154102	534,265	82.4
270_left	0.310869	0.132304	0.014551	0.141764	566,296	87.3
90_right	0	0	0	0	7658	12.2
120_right	0	0	0	0	135,137	20.8
150_right	2.502934	1.440530	0.036937	1.535981	191,471	29.5
180_right	1.422529	0.739286	0.022606	0.804200	311,443	48.0
210_right	1.326894	0.649623	0.014481	0.731533	451,772	69.7
240_right	1.371113	0.661285	0.014154	0.760515	508,636	78.4
270_right	1.029491	0.498957	0.009679	0.562762	545,708	84.1

Based on the squeezing theorem and a large number of experiments, and setting the LiDAR angle accuracy to 1°, the critical value of the relocation loss in four directions is obtained. When NDT matching points are 21.3% to 24.9% of the total, there is a high probability of missing NDT relocations. The limit is between 137,933 and 161,495. The results are shown in Table 3.

Table 3. Absolute trajectory error of critical values in four directions.

	Max (m)	Mean (m)	Min (m)	Rmse (m)	Num (m)	Ratio (%)
41_front	3.008521	1.307617	0.075811	1.509459	161,495	24.9
33_back	3.445162	1.199947	0.156978	1.278492	154,256	23.8
113_left	5.058614	0.807393	0.025104	0.913465	151,678	23.4
122_right	2.680347	1.295943	0.035288	1.380118	137,933	21.3

With an approximate number of NDT matching points, only a smaller LiDAR field of view angle is required to use the previous NDT matching points. Front NDT matching points also have larger errors. This reflects the robustness of the NDT matching points at the front and back ends.

4. Urban Dataset Testing

Tracked vehicles equipped with 64-line LiDAR are used in real environments, as shown in Figure 7. The LiDAR used here was the Ouster OS1-64, with a measurement range of 150 m, an accuracy of ± 2 cm, a vertical perspective of 30°, a horizontal perspective

of 360°, a vertical angle resolution of 0.52°, a horizontal angle resolution of 0.09°, and a rotation rate of 10 Hz. The test was conducted in a complex urban environment with obvious characteristics. As shown in Figure 8, there were curved roads, straight roads, and obvious large obstacles on the left and right sides of the crawler. The road length in the dataset was 1 s and the duration was 23 s. All evaluation experiments were run on a computer with an AMD R7-4800H processor, 16 GB RAM, and a single NVIDIA GeForce GTX1650ti GPU.



Figure 7. Tracked vehicles equipped with 64-line LiDAR.



Figure 8. High-precision maps recorded using tracked vehicles equipped with 64-line LiDAR.

It can be seen from Figures 9 and 10 that when the number of NDT matching points is less than 100, the relocalization error will increase sharply, and when the number of NDT matching points is less than 50, it is extremely easy to lose positioning. The theoretical value for the number of NDT matches per frame is set to 100. Since the number of NDT matching points is 3109, the number of NDT matching points is 249,2189. Therefore, the number of NDT valid matches cannot be less than 12.4% of the total number, and the number of

single-frame NDT valid matches cannot be less than 100. However, the complexity of the road sections (there are straight sections and curved sections) led to the fluctuation of NDT matching points in terms of the volume of the point cloud frame. According to the actual experiment, a reliable range was obtained, which was 14.7% to 16.3%, as shown in Table 4.

	Max (m)	Min (m)	Rmse (m)	Num (m)	Ratio (%)
20_front	1.715141	0.021566	0.547751	367,065	14.7
22_back	6.650557	0.018378	1.027503	378,291	15.2
97_left	2.507830	0.031737	0.995748	460,454	18.5
110_right	1.038546	0.023148	0.703556	406,214	16.3

Table 4. Absolute trajectory error of critical values in four directions.



Figure 9. Number of valid points on the front and back during NDT relocation.



Figure 10. Number of valid points on the left and right during NDT relocation.

It can be seen from Figures 9 and 10 that each group of data has four turning points with great changes. As shown in Figures 11 and 12, as the vehicle turns, the number of LiDAR-effect matching points on both sides decreases dramatically, resulting in reduced repositioning accuracy, as shown in Figures 13 and 14. The LiDAR field of view on the left and right sides becomes larger, and the number of effective matching points increases dramatically, resulting in higher repositioning accuracy, as shown in Figures 15 and 16.



Figure 11. NDT valid matching points on the direct path.



Figure 12. NDT valid matching points on curves.



Figure 13. Track error using front LiDAR data at curves.



Figure 14. Track error using back LiDAR data at curves.



Figure 15. Track error using left LiDAR data at curves.



Figure 16. Track error using right LiDAR data at curves.

As shown in Figures 17 and 18, in the normal, straight-line segment, obstacles were distributed unevenly and close by. It is easy to reduce the left and right LiDAR field

of vision, resulting in a number of NDT-effective matching points between 50 and 100, resulting in reduced relocation accuracy, or even loss of location. In the normal, straight-line segment, since most of the front and back LiDAR data were ground points, the relocation accuracy was seriously affected. The left and right LiDAR points were mostly effective feature points with rich features. As shown in Figures 19 and 20, in the straight-line segment, the positioning accuracy of the LiDAR data on the front and back sides was poor. As shown in Figures 21 and 22, the positioning effect of the LiDAR data on the left and right sides was good.



Figure 17. Road conditions with real obstacles on the left side.



Figure 18. LiDAR point clouds with real obstacles on the left.



Figure 19. Track error using front-front LiDAR data on the straight track.



Figure 20. Track error using front-back LiDAR data on the straight track.



Figure 21. Track error using front-left LiDAR data on the straight track.



Figure 22. Track error using front-right LiDAR data on the straight track.

5. Conclusions

This paper presents a LiDAR layout and field-of-view selection method based on high-precision map NDT relocation to solve the problem of large NDT relocation error and position loss when the field of view is too large to be obstructed. In order to simulate the real placement environment and obstructed LiDAR, an ROI algorithm is used to cut the LiDAR point cloud to obtain different sized LiDAR point cloud data. First, the cut point cloud data is downsampled, then relocated. The downsampling points for NDT relocation are recorded as valid matching points. The direction and angle settings for LiDAR point cloud data are optimized using RMSE values and valid matching points. The results show that in urban scenes with complex road conditions, there are more front and back matching points than left and right matching points in the unit angle. The more matching points the NDT relocation algorithm has, the higher the relocation accuracy will be. Increasing the front and back LiDAR field of view prevents the loss of repositioning. The effective matching points of single-frame LiDAR data are larger than the set threshold. By increasing the field of view of the left and right LiDAR, the repositioning accuracy can be improved. You also need to keep a safe distance from obstacles on both sides. The future research plan is to improve the NDT algorithm to speed up its processing speed, increase the positioning accuracy, and enable it to be relocated, even when the LiDAR data is sparse.

Author Contributions: Conceptualization, J.G. and L.Y.; methodology, J.G. and L.Y.; software, J.G.; validation, J.G., L.L. and L.Y.; formal analysis, J.G., F.K. and L.Y.; investigation, J.G.; resources, Y.L., J.G. and L.Y.; data curation, J.G. and L.Y.; writing—original draft preparation, J.G., L.L. and L.Y.; writing—review and editing, J.G. and L.Y.; visualization, J.G. and L.Y.; supervision, J.G., F.K. and L.Y.; project administration, J.G., H.S., J.L. and L.Y.; funding acquisition, L.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Special Funding Project for "One Event and One Discussion" for Importing Top Talent in Shandong Province (Lu Zheng Ban Zi [2018]27). This research was funded by the Zibo Unmanned Farm Research Institute Project (2019ZBXC200).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data are available upon request.

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

References

- Ng, H.F.; Hsu, L.T.; Lee, M.J.L.; Feng, J.; Naeimi, T.; Beheshti, M.; Rizzo, J.R. Real-Time Loosely Coupled 3DMA GNSS/Doppler Measurements Integration Using a Graph Optimization and Its Performance Assessments in Urban Canyons of New York. Sensors 2022, 22, 6533. [CrossRef] [PubMed]
- He, G.; Yuan, X.; Zhuang, Y.; Hu, H. An integrated GNSS/LiDAR-SLAM pose estimation framework for large-scale map building in partially GNSS-denied environments. *IEEE Trans. Instrum. Meas.* 2020, 70, 1–9. [CrossRef]
- 3. Ye, F.; Pan, S.; Gao, W.; Wang, H.; Liu, G.; Ma, C.; Wang, Y. An Improved Single-Epoch GNSS/INS Positioning Method for Urban Canyon Environment Based on Real-Time DISB Estimation. *IEEE Access* **2020**, *8*, 227566–227578. [CrossRef]
- 4. Kusaka, T.; Tanaka, T. Stateful Rotor for Continuity of Quaternion and Fast Sensor Fusion Algorithm Using 9-Axis Sensors. Sensors 2022, 22, 7989. [CrossRef]
- Li, Y.; Yang, S.; Xiu, X.; Miao, Z. A Spatiotemporal Calibration Algorithm for IMU–LiDAR Navigation System Based on Similarity of Motion Trajectories. *Sensors* 2022, 22, 7637. [CrossRef]
- Lyu, P.; Bai, S.; Lai, J.; Wang, B.; Sun, X.; Huang, K. Optimal Time Difference-Based TDCP-GPS/IMU Navigation Using Graph Optimization. *IEEE Trans. Instrum. Meas.* 2021, 70, 1–10. [CrossRef]
- Aghili, F.; Salerno, A. Driftless 3-D attitude determination and positioning of mobile robots by integration of IMU with two RTK GPSs. *IEEE ASME Trans. Mechatron.* 2011, 18, 21–31. [CrossRef]
- 8. Takai, R.; BARAWID Jr, O.; Ishii, K.; Noguchi, N. Development of crawler-type robot tractor based on GPS and IMU. *IFAC Proc. Vol.* **2010**, *43*, 151–156. [CrossRef]
- Mur-Artal, R.; Tardós, J.D. Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras. *IEEE Trans. Robot.* 2017, 33, 1255–1262. [CrossRef]
- Zhang, C.; Huang, T.; Zhang, R.; Yi, X. PLD-SLAM: A new RGB-D SLAM method with point and line features for indoor dynamic scene. *ISPRS Int. J. Geoinf* 2021, 10, 163. [CrossRef]
- Bakkay, M.C.; Arafa, M.; Zagrouba, E. Dense 3D SLAM in dynamic scenes using Kinect. In Proceedings of the Iberian Conference on Pattern Recognition and Image Analysis, Santiago de Compostela, Spain, 17–19 June 2015; Springer: Berlin/Heidelberg, Germany, 2015; pp. 121–129.
- 12. Zheng, Z.; Li, X.; Sun, Z.; Song, X. A novel visual measurement framework for land vehicle positioning based on multimodule cascaded deep neural network. *IEEE Trans. Industr. Inform.* 2020, 17, 2347–2356. [CrossRef]

- 13. Yang, K.; Zhang, W.; Li, C.; Wang, X. Accurate location in dynamic traffic environment using semantic information and probabilistic data association. *Sensors* 2022, 22, 5042. [CrossRef] [PubMed]
- 14. Zhao, Z.; Zhang, Y.; Shi, J.; Long, L.; Lu, Z. Robust Lidar-Inertial Odometry with Ground Condition Perception and Optimization Algorithm for UGV. *Sensors* 2022, 22, 7424. [CrossRef] [PubMed]
- 15. Ma, X.; Li, X.; Song, J. Point Cloud Completion Network Applied to Vehicle Data. Sensors 2022, 22, 7346. [CrossRef]
- 16. Schulte-Tigges, J.; Förster, M.; Nikolovski, G.; Reke, M.; Ferrein, A.; Kaszner, D.; Matheis, D.; Walter, T. Benchmarking of Various LiDAR Sensors for Use in Self-Driving Vehicles in Real-World Environments. *Sensors* **2022**, *22*, 7146. [CrossRef]
- 17. Wen, W.; Hsu, L.T.; Zhang, G. Performance analysis of NDT-based graph SLAM for autonomous vehicle in diverse typical driving scenarios of Hong Kong. *Sensors* 2018, *18*, 3928. [CrossRef]
- 18. Wen, W.; Bai, X.; Zhan, W.; Tomizuka, M.; Hsu, L.T. Uncertainty estimation of LiDAR matching aided by dynamic vehicle detection and high definition map. *Electron. Lett.* **2019**, *55*, 348–349. [CrossRef]
- Akai, N.; Morales, L.Y.; Takeuchi, E.; Yoshihara, Y.; Ninomiya, Y. Robust localization using 3D NDT scan matching with experimentally determined uncertainty and road marker matching. In Proceedings of the 2017 IEEE Intelligent Vehicles Symposium (IV), Los Angeles, CA, USA, 11–14 June 2017; pp. 1356–1363.
- Javanmardi, E.; Javanmardi, M.; Gu, Y.; Kamijo, S. Pre-estimating self-localization error of NDT-based map-matching from map only. *IEEE trans. Intell. Transp. Syst.* 2020, 22, 7652–7666. [CrossRef]
- Zhao, Z.; Zhang, W.; Gu, J.; Yang, J.; Huang, K. Lidar mapping optimization based on lightweight semantic segmentation. *IEEE Trans. Veh. Technol.* 2019, 4, 353–362. [CrossRef]
- 22. Zhang, J.; Singh, S. LOAM: Lidar odometry and mapping in real-time. In *Robotics: Science and Systems*; IFRR: Berkeley, CA, USA, 2014; Volume 2, pp. 1–9.
- Anderson, S.; Barfoot, T.D. RANSAC for motion-distorted 3D visual sensors. In Proceedings of the 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems, Tokyo, Japan, 3–7 November 2013; pp. 2093–2099.
- Tong, C.H.; Barfoot, T.D. Gaussian process Gauss-Newton for 3D laser-based visual odometry. In Proceedings of the 2013 IEEE International Conference on Robotics and Automation, Karlsruhe, Germany, Tokyo, Japan, 3–7 November; pp. 5204–5211.
- Zhou, Z.; Zhao, C.; Adolfsson, D.; Su, S.; Gao, Y.; Duckett, T.; Sun, L. Ndt-transformer: Large-scale 3d point cloud localisation using the normal distribution transform representation. In Proceedings of the 2021 IEEE International Conference on Robotics and Automation (ICRA), Xi'an, China, 30 May–5 June 2021; pp. 5654–5660.
- Chen, S.; Ma, H.; Jiang, C.; Zhou, B.; Xue, W.; Xiao, Z.; Li, Q. NDT-LOAM: A Real-time Lidar odometry and mapping with weighted NDT and LFA. *IEEE Sens. J.* 2021, 22, 3660–3671. [CrossRef]
- 27. Kan, Y.C.; Hsu, L.T.; Chung, E. Performance Evaluation on Map-based NDT Scan Matching Localization using Simulated Occlusion Datasets. *IEEE Sens. Lett.* **2021**, *5*, 1–4. [CrossRef]
- Wang, X.; Shang, H.; Jiang, L. Improved Point Pair Feature based Cloud Registration on Visibility and Downsampling. In Proceedings of the 2021 International Conference on Networking Systems of AI (INSAI), Shanghai, China, 19–20 November 2021; pp. 82–89.
- Yang, D.; Jiabao, B. An Optimization Method for Video Upsampling and Downsampling Using Interpolation-Dependent Image Downsampling. In Proceedings of the 2021 4th International Conference on Information Communication and Signal Processing (ICICSP), Shanghai, China, 24–26 September 2021; pp. 438–442.
- Hirose, O. Acceleration of non-rigid point set registration with downsampling and Gaussian process regression. *IEEE Trans. Pattern Anal. Mach. Intell.* 2020, 43, 2858–2865. [CrossRef] [PubMed]
- 31. Zou, B.; Qiu, H.; Lu, Y. Point cloud reduction and denoising based on optimized downsampling and bilateral filtering. *IEEE Access* 2020, *8*, 136316–136326. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.