# High precision positioning system for autopilot based on multi-sensor fusion

Luo Haoru[1,*], Liu Kechun[2]

[1]School of Automation, Wuhan University of Technology, Wuhan City, Hubei Province, China
[2]School of Automation, Southeast University, Nanjing City, Jiangsu Province, China

**Abstract.** For autonomous vehicles, autonomous positioning is a core technology in their development. A good positioning system not only helps them efficiently complete autonomous operations, but also improves safety performance. At present, the use of global positioning system (GPS) is a more mainstream positioning method, but in indoor, serious shelter and other environments, GPS signal loss will lead to positioning failure. In order to solve this problem, this paper proposes a method of mapping before positioning, and designs a set of high precision real-time positioning system by combining the technology of multi-sensor fusion. The designed system was carried on a Wuling sightseeing bus, and the mapping and positioning tests were carried out in the Nanhu Campus of Wuhan University of Technology, the East Campus of Mafangshan Campus and the underground garage where GPS signals were lost. The test results show that the system can realize the high precision real-time positioning function of the autonomous vehicle. Therefore, the in-depth study and implementation of this system is of great significance to the promotion and application of the automatic driving industry.

## 1 Introduction

At present, with the development of computer, artificial intelligence, information communication and other technologies, the trend of informationalization, interconnection and intelligence has gradually penetrated into all walks of life, bringing more convenient, environmentally friendly and intelligent equipment and facilities [1]. They are rapidly changing people's life, food, clothing, housing and transportation. Among them, automatic driving technology is a dazzling pearl, which has been widely valued and applied in military, aviation, aerospace, industry, transportation, medical treatment, life and other fields [2]. In addition, after the epidemic, consumers' demand for non-contact or low-contact services will continue. Based on this change, the development of the automatic driving industry will become a larger trend [3].

Positioning technology is the core of automatic driving technology, which is the premise of autonomous operation of autonomous driving vehicles, and also the core of automatic driving technology. The most important significance of high precision positioning is to make automatic driving safer. Only with high enough precision positioning can the vehicle plan a relatively safe route and improve the safety of automatic driving [4]. In addition, autonomous driving requires safety all the time. Therefore, the positioning function of high-level automatic driving requires not only high accuracy, but also higher frequency updates. How to achieve real-time positioning of automatic driving is also a very important topic.

Common positioning technology is divided into outdoor positioning and indoor positioning two kinds, outdoor positioning has base station positioning, GPS positioning, etc. Indoor positioning includes BLE Bluetooth wireless positioning technology, RFID radio frequency identification technology, Wi-Fi positioning technology and other methods [5]. At present, most autonomous driving vehicles complete positioning through differential GPS technology, which can solve the problem of positioning accuracy, but cannot solve the problem of occlusion and reflection [6]. In a complex environment, GPS signals are easily affected greatly, and in severe cases, signals are easy to be weakened or even lost, leading to positioning failure [7]. In a high-rises environment, for example, GPS receivers around tall buildings can easily lose all satellite signals in one direction. In this case, the positioning accuracy of GPS is difficult to meet the requirements of automatic driving. In addition, in the intelligent logistics of the factory area, because the logistics vehicles have to shuttle indoors and outdoors in many factories, the GPS signal is weak indoors and easy to be lost, so that the logistics vehicles lose some GPS signals in the operation process. Similarly, in the tunnel, underground garage and other scenarios, it is also easy to cause GPS signal lock loss. When the GPS-signal-based autonomous driving car encounters the above situation, the problem of inaccurate positioning will occur. At the least, the ride experience of the passengers will be affected; at the worst, the

---

* Corresponding author: lhr-_.289753@whut.edu.cn
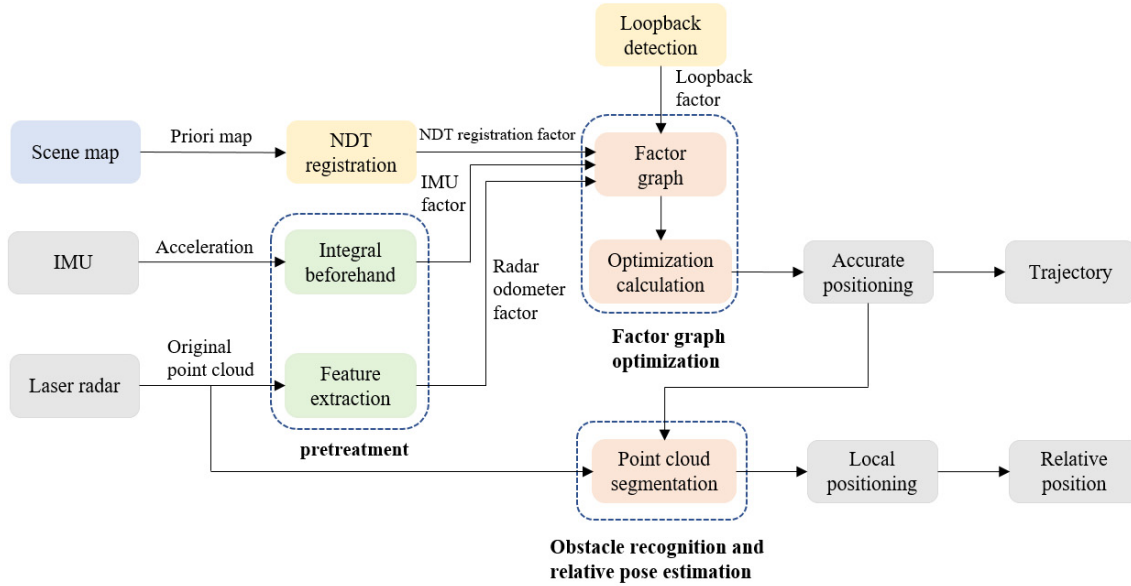  213181824@seu.edu.cn

**Fig. 1.** Overall structure diagram of the system

vehicle will run off the road or drive in the wrong direction, resulting in traffic accidents, with unimaginable consequences [8].

Synchronous localization and map construction (SLAM), which has emerged in recent years, is the key technology to solve the above problems [9]. As the core technology of robot positioning and navigation, SLAM is attracting increasing attention in the industry. In the field of robotics, SLAM has been successfully applied. But the SLAM algorithm used for robots is designed for highly structured, well-lit environments indoors, not for variable lighting outdoors. At the same time, autonomous vehicles can run faster than robots, which requires faster and more efficient algorithms. According to the data, when pure SLAM technology is applied to the autonomous driving vehicle for testing, the absolute error is relatively large, with an average of about 0.5m [10]. The influence of such errors on the positioning accuracy of the autonomous vehicle navigation task is not acceptable. In other words, the current SLAM algorithm of autonomous driving vehicles needs to be more accurate and faster [11].

In view of the above reasons, through the fusion of multi-sensor signals, the system adopts the technical method of mapping before positioning to solve the problem of high-precision real-time positioning of automatic driving vehicles in the absence of GPS signals, and popularizes and applies it to all kinds of scenes requiring high-precision positioning.

## 2 Materials and Methods

In this paper, a factor graph optimization scheme based on various sensors such as lidar, IMU and GPS is proposed to optimize laser SLAM by the factor graph method and construct a high-precision point cloud map. After the construction of the point cloud map is completed, the NDT registration algorithm is used to determine the optimal match between the two point

clouds [12]. IMU pre-integration factor, lidar range factor, NDT registration factor and closed-loop factor were introduced, and the precise positioning function of the system was realized through optimization of back-end factor graph. At the same time, the point cloud segmentation method is used to realize the recognition of obstacles and the relative pose estimation to complete the relative positioning. The overall structural framework of the system is shown in Fig. 1.

### 2.1. IMU Pre-integration

Firstly, the coordinate system and symbols used in this system are defined. The world coordinate system is represented as *w*, and the object principal coordinate system is represented as *b*. Meanwhile, for convenience, it is assumed that the IMU coordinate system coincides with the object coordinate system. Object state *X* can be written as:

$$X = \left[ R^T, p^T, v^T, b^T \right]^T \quad (1)$$

Where, $R \in SO(3)$ is the rotation matrix, $p \in \mathbb{R}^3$ is the position vector, *v* is the velocity, and *b* is the IMU deviation. The change from *b* to *w* is denoted by $\mathbf{T} = \begin{bmatrix} \mathbf{R} & | p \end{bmatrix}$.

The kinematics formula of IMU in the discrete state can be obtained as follows:

$$p_{wb_{k+1}} = p_{wb_k} + v_k^w \Delta t + \frac{1}{2} a \Delta t^2$$
$$v_{k+1}^w = v_k^w + a \Delta t \quad (2)$$
$$q_{wb_{k+1}} = q_{wb_k} \otimes \begin{bmatrix} 1 \\ \frac{1}{2} \omega \delta t \end{bmatrix}$$

Where, $a = q_{wb_k} \left( a^{b_k} - b_k^a \right) - g^w$, $\omega = \omega^{b_k} - b_k^g$.

The average IMU device's data update frequency is between 100Hz and 1000Hz, while the lidar's data update frequency is generally around 10Hz, which is

much higher than the lidar's data refresh frequency. As shown in Fig. 2, the current timestamp is *i* and the point cloud timestamp of the next frame is *j*, so measurement data of several IMU can still be received before the point cloud data is received in the next frame. The IMU data in this period of time are pre-integrated, and the results are used for pose estimation and distortion elimination of the point cloud data in the next frame. Therefore, the IMU product component between the point cloud data of the upper and lower frames needs to be solved [13].
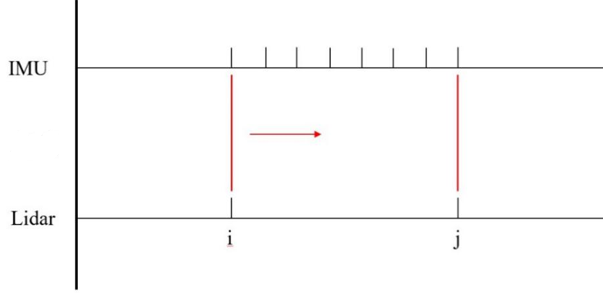


**Fig. 2.** Schematic diagram of sensor time series data

The motion model in Equation (2) can be transformed into a pre-integral model through Equation (3):

$$q_{wb_t} = q_{wb_i} \otimes q_{b_i b_t} \qquad (3)$$

According to Equation (2), the IMU data of $\Delta t \in [i, j]$ within a period of time are directly integrated to obtain the pre-integration component:

$$\alpha_{b_i b_j} = \iint_{t \in [i,j]} (q_{b_i b_t} a^{b_t}) \delta t^2$$

$$\beta_{b_i b_j} = \int_{t \in [i,j]} (q_{b_i b_t} a^{b_t}) \delta t \qquad (4)$$

$$q_{b_i b_j} = \int_{t \in [i,j]} q_{b_i b_t} \otimes \begin{bmatrix} 0 \\ \frac{1}{2} \omega^{b_t} \end{bmatrix} \delta t$$

The integral formula of PVQ after rearrangement is as follows:

$$\begin{bmatrix} p_{wb_j} \\ v_j^w \\ q_{wb_j} \\ b_j^a \\ b_j^g \end{bmatrix} = \begin{bmatrix} p_{wb_i} + v_i^w \Delta t - \frac{1}{2} g^w \Delta t^2 + q_{wb_i} \alpha_{b_i b_j} \\ v_i^w - g^w \Delta t + q_{wb_i} \beta_{b_i b_j} \\ q_{wb_i} q_{b_i b_j} \\ b_i^a \\ b_i^g \end{bmatrix} \qquad (5)$$

## 2.2 Lidar Odometer

Due to the large number of lidar point clouds, direct processing will be a waste of computing resources. In addition, the point cloud is directly registered iteratively. In the case of no obvious difference between each point cloud, it is easy to mismatch the point cloud or fall into local optimum, and the overall point cloud cannot be further registered to reach global optimum. Therefore, it is necessary to extract the features of point clouds. Point cloud clusters with obvious features in point clouds are

marked first, and the eigenvalues are calculated. Then, laser point cloud registration is performed based on feature point cloud clusters to reduce the amount of calculation of point cloud registration and the probability of mismatching of laser point cloud [14].

(1) Feature point extraction

For Ouster OS1 16-line lidar used in this system, feature extraction is carried out for each line according to curvature. The specific calculation method is as follows:

1) Point *i* is defined as a random dot cloud at time *k*, and the nearest ten points in the neighborhood of point *i* are taken as the neighborhood point set *S*; *j* is defined as the point cloud that does not coincide with point *i* in the point set;

2) According to the point cloud coordinate information, the coordinate of point *i* at time *k* in the lidar coordinate system is defined as $X_{(k,i)}^L$, similarly, the coordinate of inner point *j* in the neighborhood point set can be expressed as $X_{(k,i)}^L$;

3) The average curvature *c* corresponding to point *i* can be calculated by using Equation (8) as follows:

$$c = \frac{1}{|s| \cdot \left\| X_{(k,i)}^L \right\|} \left\| \sum_{j \in S, j \neq i} (X_{(k,i)}^L - X_{(k,j)}^L) \right\| \qquad (6)$$

4) In order to make the feature point cloud evenly distributed, each radar laser line is evenly divided into 6 parts (6 *sub*).

5) Sort the points in each *sub* according to the curvature, set the curvature threshold *k*, the points with curvature greater than *k* are angular feature points, and the points with curvature less than *k* are surface feature points, and obtain the preliminary feature point cloud.

(2) Feature point matching

1) Establish the feature line: take the point cloud feature point *i* as an example at time *t*, use k-d Tree algorithm to find the nearest point *j* at time *t* of point *i*, and find the sub-nearest point *l* centered on *j*, and take point $(j, l)$ as the corresponding point cloud at time *t*.

2) Establish the feature surface: similar to (1), after finding the point *j*, *l* near the feature point *i* at time *t*, take point *l* as the center and use k-d Tree to find the sub-nearest point *m*, and take $(j, l, m)$ as the corresponding point cloud at time *t*.

3) Constraint equations of characteristic line and characteristic surface are respectively established as follows:

$$d_\varepsilon = \frac{\left| (\tilde{X}_{(k,i)}^L - \bar{X}_{(k-1,j)}^L) \times (\tilde{X}_{(k,i)}^L - \bar{X}_{(k-1,l)}^L) \right|}{\left| \bar{X}_{(k,i)}^L - \bar{X}_{(k-1,j)}^L \right|} \qquad (7)$$

$$d_H = \frac{(\tilde{X}_{(k,i)}^L - \bar{X}_{(k-1,j)}^L) \bullet ((\bar{X}_{(k-1,j)}^L - \bar{X}_{(k-1,l)}^L) \times (\bar{X}_{(k-1,j)}^L - \bar{X}_{(k-1,m)}^L))}{(\bar{X}_{(k-1,j)}^L - \bar{X}_{(k-1,l)}^L) \times (\bar{X}_{(k-1,j)}^L - \bar{X}_{(k-1,m)}^L)} \qquad (8)$$

After the distance is calculated, if the distance is small enough to meet the conditions, the corresponding relationship is preserved.

## 2.3 Backend optimization based on factor graph

The system receives sensor data from 3D lidar, IMU and optional GPS. By observing the data from these sensors, the state and trajectory of the object can be estimated. This state estimation problem can be called the maximum a posteriori estimation problem. Since the inference of factor graph is more suitable than the inference of Bayesian network, a factor graph is chosen to model this problem. Meanwhile, under the assumption of Gaussian noise model, the MAP corollary of the problem is equivalent to solving a nonlinear least squares problem [15]. Moreover, without loss of generality, the system can further combine the measurement data of other sensors, such as altitude measured by an altimeter or direction obtained from a compass, to complete the state and trajectory prediction of objects.

For the construction of a factor graph, four factors and one variable are introduced. Where, this variable belongs to the node in the factor graph, and it represents the state of the object at a certain time. The above four types of factors are: (a) IMU pre-integration factor, (b) lidar odometer factor, (c)NDT registration factor and (d) loop back factor. When the pose of the object changes beyond the user-defined threshold, a new state node X is added to the graph. On this basis, the factor graph can be optimized by using ISAM2 algorithm [16]. The processes that produce these factors will be described in other sections.

## 2.4 Obstacle identification and relative pose estimation

Accurate positioning in real time is crucial for autonomous driving technology. Therefore, in order to ensure the safety of autonomous driving vehicles all the time, the local positioning technology of obstacle identification and relative pose estimation is also an important issue in our mapping and positioning system.

The purpose of obstacle recognition and relative pose estimation is to segment the point clouds of obstacles distributed separately in the point cloud coordinate system of a single frame driving field scenic spot cloud one by one and extract the information of obstacles that can be understood and utilized by the autonomous driving vehicle. For lidar point cloud, the separation of measuring points of different obstacles is obvious. Clustering algorithm can usually be used to divide point cloud data and separate single obstacle point cloud one by one, so as to further extract obstacle object information.

(1) Obstacle recognition based on Euclidean algorithm

The basic idea of Euclidean clustering is to take the Euclidean distance between the elements in the data set as the criteria for proximity discrimination. If the Euclidean distance between the measuring points is less than a certain preset threshold, it is considered to be the measuring point of the same obstacle.

For any two measuring points $p_i\left(x_i, y_i, z_i, l_i\right)$ and $p_j\left(x_j, y_j, z_j, l_j\right)$, the Euclidean distance can be calculated by the following formula:

$$D = \sqrt{\left(x_i - x_j\right)^2 + \left(y_i - y_j\right)^2 + \left(z_i - z_j\right)^2} \qquad (9)$$

For cloud data P of scenic spots in driving field in any given single frame of lidar, the basic process of neighborhood growth clustering based on Euclidian distance is shown in Fig. 3.
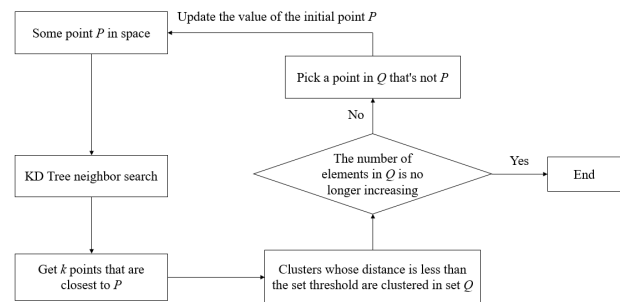


**Fig. 3.** Euclidean clustering flow chart

(2) Relative pose estimation of obstacles based on PCA

PCA principal component analysis (PCA) is an important statistical analysis method. It carries out linear transformation on the data set of n-dimensional features that are related to each other, that is, it transforms the data set from the original coordinate space to the new coordinate space through translation and rotation, so that the data can be represented by the features after rotation. The specific process of establishing OBB directed bounding box of single obstacle cluster $C_k$ based on PCA principal component analysis is as follows:

1) Combine data according to coordinate type;

2) Calculate the mean value of the data set and construct the covariance matrix;

3) Then the eigenvalue and eigenvector are solved according to the covariance matrix, where the larger eigenvalue is the direction of the OBB bounding box;

4) The characteristic vector obtained is the new coordinate system, and the original data can be lowered to this coordinate system to obtain the length, width and center point of the OBB bounding box.

Thus, the OBB directed bounding box of single obstacle clustering was established based on PCA principal component analysis method, and the basic spatial information of obstacles was extracted to complete the spatial information description of obstacles.

## 2.5 Loop detection

Loop detection is a very key part of SLAM, and of course, it is also one of the core technologies for the construction of high-precision map of autonomous driving. Loop detection means that the unmanned system can identify the closed loop formed in the mapping process of SLAM, so as to optimize the accumulated errors from observation. Therefore, the SLAM map can be accurately connected at the "stitching point" of the

loop, and the repeated measurements in the same section can be accurately fitted. Loop detection is very necessary for map construction of large area and large scene in this system.

The flow chart of the whole loop detection link of this system is shown in Fig. 4.
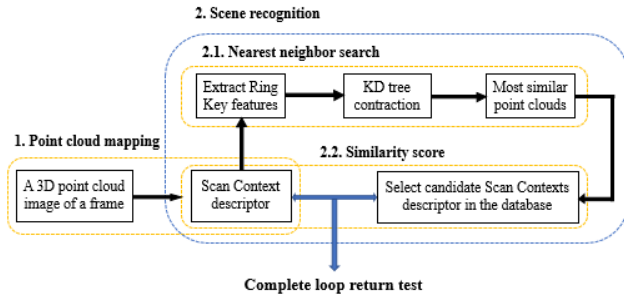


**Fig. 4.** Flow chart of loop detection

## 2.6 High-precision positioning based on NDT

Now really SLAM to realize autonomous positioning system is very difficult, because for the moment pure SLAM cannot reach to the requirement of positioning accuracy, reliability, automatic driving now study hard to realize the online mapping and positioning of unmanned systems [17]. Therefore, this work chooses to use the NDT registration algorithm to do the positioning of the automatic driving system on the premise of having a high-precision map, so as to realize the positioning function is more realistic and simple.

(1) Point cloud filtering based on VoxelGrid down-sampling method

Automatic driving positioning has high requirements for real-time performance, and the less time it takes for point cloud registration, the better. Therefore, the input point cloud can be de-sampled to improve the speed of NDT registration, and the VoxelGrid de-sampled method can be used to reduce the density of the input point cloud.

The principle of VoxelGrid down-sampling implemented by PCL is to create a series of three-dimensional voxel grids through the input point cloud data, which is similar to a collection of tiny three-dimensional cubes in space. Then, in each voxel, the barycenter of all points in the VoxelGrid is used to display other points in the voxel approximately. The specific process is shown in Fig. 5.
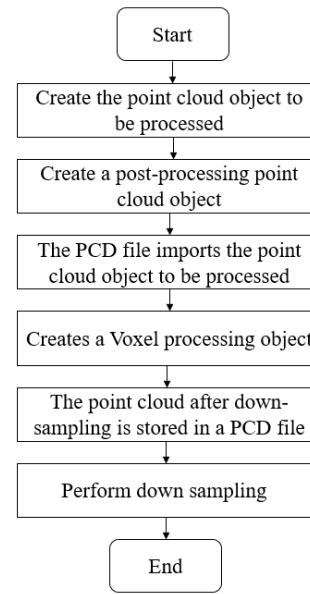


**Fig. 5.** Flow Chart of VoxelGrid Down-sampling Method

(2) High precision positioning based on NDT registration algorithm

The basic idea of NDT algorithm is to convert the reference point cloud (high-precision map) into the normal distribution of multi-dimensional variables and then carry out matching. The core idea of this transformation of the NDT is to map the point cloud to a smooth surface represented by a set of local probability density functions (PDF), each of which describes the shape of a part of the surface. If the transformation parameters can make the two laser data match well, then the probability density of the transformation point in the reference frame will be very large. Therefore, the transformation parameter that maximizes the sum of probability densities can be obtained by using the optimization method, and then the two laser-point cloud data will match best [18]. The specific steps of NDT algorithm are as follows:

1) Divide the space occupied by the reference point cloud into grids or voxels of specified size, and calculate the multi-dimensional normal distribution parameters of each grid:

$$q = \frac{1}{n}\sum_i x_i \tag{10}$$

$$\Sigma = \frac{1}{n}\sum_i (x_i - q)(x_i - q)^T \tag{11}$$

Where, $q$ is the mean value; $\Sigma$ is the covariance matrix; $x_i$ represents all scan points in a grid.

2) Set initialization transformation parameters, that is, the pose to be estimated:

$$p = \left(t_x, t_y, \phi\right)^T \tag{12}$$

3) For the point cloud that needs registration, transform it into the grid of the reference point cloud through transformation $T$:

$$T : \begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} \cos\phi & -\sin\phi \\ \sin\phi & \cos\phi \end{pmatrix}\begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} t_x \\ t_y \end{pmatrix} \tag{13}$$

4) Calculate the probability density of each conversion point according to the normal distribution parameters:

$$p(x) \sim \exp\left(-\frac{(x-q)^T \sum^{-1}(x-q)}{2}\right) \quad (14)$$

5) NDT registration score is obtained by adding the calculated probability densities of each grid:

$$score(p) = \sum_i \exp\left(-\frac{(x_i^{'}-q_i)^T \sum_i^{-1}(x_i^{'}-q_i)}{2}\right) \quad (15)$$

(6) Newton optimization algorithm is used to optimize the objective function $-score$, that is, to find the transformation parameter $p$ to maximize the value of $score$.

## 3 Results & Discussion

This test is divided into two parts: large-scale drawing construction and vehicle positioning. Data collection and testing are carried out on the Wuling sightseeing vehicle as the carrying platform, as shown in Fig. 6. In order to avoid the interference of dynamic objects to the algorithm, data collection was carried out at a time when there were few people and vehicles. The vehicle speed was kept at about 20km/h. Multiple data collection was carried out to build a large-scale point cloud map. The vehicle positioning test was carried out on the established high-precision map. In order to verify the positioning accuracy of the system, the value of differential GPS was taken as the reference value in the same scene, and the positioning result was compared with the trajectory of differential GPS.



**Fig. 6.** Test Platform -- Wuling Sightseeing Car

### 3.1 Large-scale drawing test

Team members designed a series of experiments to qualitatively and quantitatively analyze the proposed framework. The sensor suite used in this system includes Ouster 32-wire lidar, IMU and differential GPS as signal input to the system. In addition, differential GPS is used as the control group to provide real values for the verification of system performance. In order to ensure the universality of the experimental results, the tests were carried out in Nanhu Campus of Wuhan University of Technology, East Campus of Mafangshan Campus of Wuhan University of Technology and underground garage respectively. The detailed information of these data sets is shown in Table 1.

**Table 1.** Dataset details

| Serial Number | Radar Frames | Elevation Changes(m) | Total Mileage(m) |
| --- | --- | --- | --- |
| Nanhu | 2476 | 8 | 1800 |
| Mafangshan | 1859 | 9 | 1650 |
| Garage | 2219 | 8 | 750 |

**A. Wuhan University of Technology Nanhu Campus**

To collect the data set, test subjects drove a Wuling sightseeing car along the main road of Wuhan University of Technology's Nanhu campus and returned to the same location six minutes later. This test included loop paths to provide loop factors to the system for global consistency optimization. The GPS data received was only used during initialization to compute the vehicle attitude and align the vehicle coordinate system to the world coordinate system. The results of drawing construction using this system are shown in Fig.7.
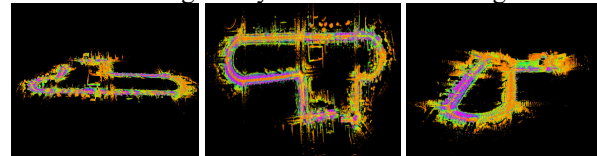


**Fig. 7.** (left) main view (middle) top view (right) side view

**B. Wuhan University of Technology, Ma Fangshan Campus East Campus**

The test subjects drove a Wuling sightseeing car to collect a set of data in the East Campus of Wuhan University of Technology's Mfangshan Campus, where buildings, trees and other shielding objects are less, so GPS signals can be obtained more accurately and multi-stage loop detection is convenient at the same time. After data collection, LIO-SAM algorithm was used to build the map, as shown in Fig. 8.
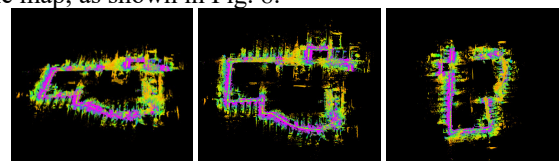


**Fig. 8.** (left) main view (middle) top view (right) side view

**C. Underground parking**

After completing the first two groups of tests, the team members further used the system to construct the point cloud map in the underground parking lot where the GPS signal was lost. The construction results are shown in Fig. 9.
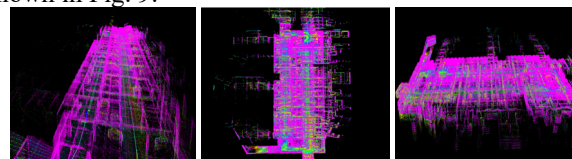


**Fig. 9.** (left) main view (middle) top view (right) side view

It can be seen that the system can construct a high-precision map with rich information even when the GPS signal is completely lost.

### 3.2 Vehicle positioning test

The team uses the NDT positioning algorithm to achieve real-time positioning on the established high-precision map. The positioning process is shown in Fig. 10.
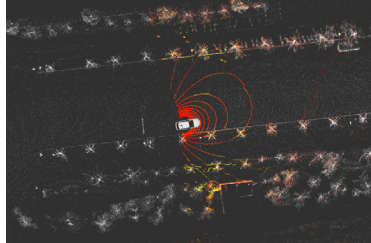


**Fig. 10.** Autonomous positioning process of the system

In order to verify the superiority of the positioning accuracy of this system, the value of differential GPS was taken as the control value in the same scene, and the positioning result was compared with the trajectory of differential GPS. The positioning trajectory in scenes A and B is shown in Fig. 11.



**Fig. 11.**
(left) Positioning trajectory under the satellite map of scene A
(right) Positioning trajectory under the satellite map of scene B

It can be seen from the above figure that the trajectory of NDT algorithm is stable and accurate. According to the calculation, the horizontal error and rotation error of NDT algorithm have been kept in a small range. On the whole, the high precision pose estimation based on NDT algorithm shows good performance, and the positioning trajectory with high accuracy can be obtained.

In addition, the mean square error of the key frame pose and the differential GPS output pose is chosen as the expression quantity of the mapping accuracy. The MSE of the two data sets of Wuhan University of Technology Nanhu Campus and the East Campus of Mafangshan Campus are calculated respectively, namely, the accuracy of positioning. The results are shown in Table 2:

**Table 2.** MSE of key frame pose in the process of positioning

| Scenario | Mean Square Error(m) |
|---|---|
| Nanhu | 0.14 |
| Mafangshan | 0.10 |

## 4 Conclusions

In this paper, based on SLAM technology, a mapping first and localization second method is proposed. Combined with multi-sensor fusion technology, this method solves the problem that the automatic driving system cannot realize accurate localization when the GPS signal is weakened or lost. The method was verified on a Wuling sightseeing bus, and the large-scale mapping and positioning functions of the system were tested. Through the calculation and analysis of the test results, it is successfully proved that it can accomplish the high precision real-time positioning function without GPS signal, which reaches the experimental expectation.

## References

1. Zhang Shunqing. Tamping the Digital Base of Intelligent Road and Building a New System of Vehicle-to-road Cooperation [J]. Zhangjiang Science and Technology Review,2021(01):32-35.

2. Bai Chuan. The Dawn of Autonomous Driving Business Model [J]. Automotive Engineer,2021(Z1):35-39.

3. Cui Xiaosu. Reconstruction of Industry Value Chain by "New Four Modernizations" [J]. Decision Exploration (I),2021(02):38-39.

4. Cyber Policy Institute, SADI think tank. Intelligent and Connected Vehicle Policy Analysis [N]. China Computer Journal,2021-02-01(012).

5. Zhao Yuhui. Autonomous driving may usher in a good development opportunity [N]. China Finance and Economics News,2021-02-04(005).

6. Luo Yaping, Shang Yongqiang, Liu Zhibo, Zhao Gengyun, Li Wanmin, Lin Xiaojun. Design of unmanned vehicle navigation system based on multi-sensor information fusion [J]. Journal of Lanzhou Institute of Technology,2020,27(02):71-76.

7. Lu Biao, Chen Yu, Qiu Huili. Research and Design of Multi-sensor Information Fusion and Information Acquisition System Based on Wireless Transmission Technology [J]. Journal of Lanzhou University of Arts and Sciences (Natural Science Edition),2019,33(04):48-51.

8. Zhang J, Singh S. LOAM: Lidar Odometry and Mapping in Real-time[C]// Robotics: Science and Systems Conference. 2014.

9. SMITH R C, CHEESEMAN P. On the representation and estimation of spatial uncertainty[J]. The international journal of Robotics Research, 1986, 5(4): 56-68.

10. LENAC K, KITANO A, CUPEC R, et al. Fast Planar Surface 3D SLAM Using LIDAR[J]. Robotics and Autonomous Systems, 2017, 92: 197-220.

11. PIERZCHALA M, GIGUERE P, ASTRUP R. Mapping forests using an unmanned ground vehicle with 3D Lidar and graph-SLAM[J]. Computer and Electronics in Agriculture, 2018, 145.

12. Dai Kai, Shen Qiren, Wu Guanpu, Zhao Yang, Ma Fangwu. Automotive Abstraction,2021(02):1-8. A

Review of LiDAR Based SLAM and Fusion Localization [J]. Automobile Abstraction,2021(02):1-8.

13. An Jingya. Development of High Precision Positioning and Environment Construction Algorithm for Autonomous Driving in Park [D]. Jilin University,2020.

14. Pang Chenglin. Research on LiDAR-IMU Tight-Coupled SLAM Algorithm Based on Local Map Joint Optimization [D]. Shandong University,2020.

15. F. Dellaert and M. Kaess, "Factor Graphs for Robot Perception," Foundations and Trends in Robotics, vol. 6(1-2): 1-139, 2017.

16. M. Kaess, H. Johannsson, R. Roberts, V. Ila, J.J. Leonard, and F. Dellaert, "iSAM2: Incremental Smoothing and Mapping Using the Bayes Tree," The International Journal of Robotics Research, vol. 31(2): 216-235, 2012.

17. Z. Xia and S. Tang, "Robust self-localization system based on multi-sensor information fusion in city environments," 2019 International Conference on Information Technology and Computer Application (ITCA), Guangzhou, China, 2019, pp.

18. Huang F, Yang H, Tan X, Peng S, Tao J, Peng S. Fast Reconstruction of 3D Point Cloud Model Using Visual SLAM on Embedded UAV Development Platform. Remote Sensing. 2020; 12(20):3308.