

3D Point Cloud Generation Based on Multi-Sensor Fusion

Yulong Han ^{1,†}, Haili Sun ^{1,*}, Yue Lu ^{1,†}, Ruofei Zhong ¹, Changqi Ji ¹ and Si Xie ²

¹ Beijing Advanced Innovation Center for Imaging Theory and Technology, Key Laboratory of 3D Information Acquisition and Application, MOE, College of Resource Environment and Tourism, Academy for Multidisciplinary Studies, Capital Normal University, Beijing 100048, China

² State Key Laboratory of Rail Transit Engineering Informatization (FSDI), Xi'an 710043, China

* Correspondence: sunhaili@cnu.edu.cn

† These authors contributed equally to this work.

Abstract: Traditional precise engineering surveys adopt manual static, discrete observation, which cannot meet the dynamic, continuous, high-precision and holographic fine measurements required for large-scale infrastructure construction, operation and maintenance, where mobile laser scanning technology is becoming popular. However, in environments without GNSS signals, it is difficult to use mobile laser scanning technology to obtain 3D data. We fused a scanner with an inertial navigation system, odometer and inclinometer to establish and track mobile laser measurement systems. The control point constraints and Rauch-Tung-Striebel filter smoothing were fused, and a 3D point cloud generation method based on multi-sensor fusion was proposed. We verified the method based on the experimental data; the average deviation of positioning errors in the horizontal and elevation directions were 0.04 m and 0.037 m, respectively. Compared with the stop-and-go mode of the Amberg GRP series trolley, this method greatly improved scanning efficiency; compared with the method of generating a point cloud in an absolute coordinate system based on tunnel design data conversion, this method improved data accuracy. It effectively avoided the deformation of the tunnel, the sharp increase of errors and more accurately and quickly processed the tunnel point cloud data. This method provided better data support for subsequent tunnel analysis such as 3D display, as-built surveying and disease system management of rail transit tunnels.

Keywords: mobile laser scanning; 3D point cloud; inertial navigation; RTS smoothing; recursive average filter



Citation: Han, Y.; Sun, H.; Lu, Y.; Zhong, R.; Ji, C.; Xie, S. 3D Point Cloud Generation Based on Multi-Sensor Fusion. *Appl. Sci.* **2022**, *12*, 9433. <https://doi.org/10.3390/app12199433>

Academic Editors: Mingliang Zhou and Dongming Zhang

Received: 16 August 2022

Accepted: 19 September 2022

Published: 20 September 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 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/>).

1. Introduction

With its fast, accurate and convenient measurement advantages, mobile laser scanning technology is widely used in areas such as unmanned driving, 3D map construction, deformation monitoring, smart city construction, tunnel maintenance and measurement tasks [1–4], and rail transit with high density and short windows at train stops. The most common method to obtain 3D point cloud data in mobile laser scanning is to combine laser scanners with global navigation satellite system (GNSS) and inertial measurement units (IMUs) to collect data through time synchronization and calibration of unified spatial coordinates and fuse them to generate 3D point cloud data in an absolute coordinate system. [5,6].

Since there is no GNSS signal in the tunnel scene, the inertial navigation system positioning and attitude errors rapidly accumulate over time and quickly increase to the tolerance limit of the measurement [7]. To this end, some research institutions have developed a tunnel mobile detection system based on relative scanning, which integrates a laser scanner, a rail inspection car and an odometer. The three-dimensional data of the tunnel are obtained by the two-dimensional section obtained by the helical scanning of the laser scanner according to the mileage [8]. The acquired tunnel data does not have true three-dimensional coordinates and true alignment since the system does not have attitude

and position measurement sensors such as inertial navigation. To obtain 3D data of scenes without satellite signals, such as tunnels, some research teams use auxiliary equipment to correct coordinates and obtain point clouds.

The Amberg GRP series track measurement system [9] is equipped with a reflective prism on a trolley body. During data acquisition, the total station measures the absolute three-dimensional coordinates of the prism center at intervals to correct the trajectory. The maximum deviation of the point cloud coordinates is 30.3 mm, and the average deviation is 10.4 mm. Despite the high accuracy, the measurement efficiency of this stop-and-go mode is low. In 2017, Mei Wensheng et al. studied a method of fixing a reflector on the tunnel wall or CPIII control pile, correcting the mileage through the target center, and then combining the design data to restore the three-dimensional point cloud in the absolute coordinate system. Based on the three-dimensional transformation matrix, Du Liming and Han Yulong et al. studied the unified positioning of the cross-section related to coordinate transformation based on the relative measurement mode and tunnel design data [10,11]. The points in the coordinate system are converted into the design route coordinate system, and the point cloud data obtained by the relative measurement technology are processed through the three-dimensional coordinate conversion and various transformation parameters to realize the unified positioning and attitude correction of the scanning section and obtain the absolute point cloud data. However, the problem with this method is that the design data may not match the actual construction, or obtaining the design data is difficult due to the track's age. In our previous study [12], we proposed methods such as fitting the design data based on the measured data and correcting the trajectory based on the segmented control points of the tunnel to generate 3D point clouds. However, these methods also have low absolute accuracy, cannot solve inertial navigation drift, and are only suitable for specific tunnels and other problems.

In indoor positioning, simultaneous localization and mapping (SLAM) are more commonly used in 3D point cloud reconstruction. Since GNSS signals are not required, the pose estimation can be performed in closed spaces by means of the matching algorithm before and after the point cloud to achieve fast and accurate positioning. SLAM has also been widely used in the complete measurement of urban rail transit projects and the positioning and modeling of underground space [13,14]. However, due to the few features and strong repeatability of the tunnel scene, it is difficult to complete the registration of point clouds between the front and rear frames, resulting in low positioning accuracy.

A possible solution for the above problems and to obtain high-precision 3D point cloud data with absolute coordinates is to use control points combined with inertial navigation to obtain high-precision trajectory data. The rail transit scene is limited by the short skylight time and high operating pressure, and the control points cannot be arranged in a large number and densely distributed. Therefore, it is necessary to make full use of the observation data of the control points to constrain the inertial navigation. This is a typical fixed interval filtering and smoothing problem. The fixed interval smoothing focuses on the optimal estimation of the state of a fixed measurement interval, and all the observations in the measurement interval can be used to estimate the state at any time in the interval.

The two main types of fixed interval smoothing algorithms are the two-filter smoothing algorithm [15–17] and Rauch-Tung-Striebel (RTS) smoothing algorithm [18]. The two-filter smoothing algorithm includes two independent filtering processes, which makes it more complicated to implement and requires storing more variables. The RTS smoothing algorithm combines the reverse filter and the smoothing process into a single reverse recursive form, including a forward filtering process and the reverse smoothing process. The structure is relatively simple, the implementation is convenient, and the calculation efficiency is relatively high. It is more extensively used in engineering [19,20]. Jiang [21] used the RTS smoothing algorithm for track smoothness detection and proposed the concept of dimensionality reduction using the Kalman filter in the vertical and horizontal directions. We draw on the above concepts and apply the RTS smoothing algorithm to

solve the problem of constrained inertial navigation and odometry to solve the trajectory in a non-GNSS scenario.

For the convenience of expression, we took the high-precision laser scanning sensor as the main body and integrated multi-type sensors (inertial navigation system, odometer, inclination sensor, gauge sensor, etc.) as the track mobile laser measurement system. The scanner was integrated with the inertial navigation system, odometer and inclinometer to establish and track the mobile laser measurement system, and the control points were integrated to establish inertial navigation and odometer constraints. Combined with the RTS smoothing algorithm, a 3D point cloud generation method based on multi-sensor fusion was established. The error between track center points was calculated according to the inertial navigation trajectory and the control point coordinates. The mileage and inertial navigation attitude angle were corrected by the RTS smoothing algorithm to obtain high-precision trajectory data; high precision was used for the inertial navigation roll angle. The static inclinometer was used to correct the inclination, and then the three-axis attitude angle was optimized by the recursive average filtering algorithm to suppress the jitter noise generated by the high-frequency acquisition. Finally, the corrected data were used to generate high-precision three-dimensional point cloud data in the absolute coordinate system, which effectively avoided the sharp increase of deformation and errors in the tunnel-moving scanning-point cloud. The 3D point cloud data could be used for tunnel leakage detection, intrusion detection, completion survey, disease system management and other follow-up tunnel analyses. They also provided good data support for applications such as rail transit three-dimensional scene reconstruction, intelligent railway construction and track measurement.

2. Filtering and Smoothing Theory

We proposed a method of three-dimensional point cloud generation based on multi-sensor fusion for the rapid accumulation of inertial navigation positioning and attitude determination errors in tunnels and other scenes without GNSS signals. This method used a few control points, constrained inertial navigation and odometer data through RTS smoothing algorithm, and generated three-dimensional point cloud data by fusion. In this section, we first analyze the inertial navigation position update and error equation and introduce the RTS smoothing algorithm.

2.1. Subsection Location Update Algorithm

The attitude update adopted the quaternion algorithm, and the quaternion algorithm is expressed as follows:

$$Q(q_0, q_1, q_2, q_3) = q_0 + q_1i + q_2j + q_3k \quad (1)$$

where q_0, q_1, q_2, q_3 are real numbers, i, j and k are mutually orthogonal unit vectors and imaginary units $\sqrt{-1}$. According to the quaternion algorithm, the position can be updated as

$$r^R = C_{ib}^R r^b \quad (2)$$

The R system is the coordinate navigation system, and the local geographic coordinate system is used as the coordinate navigation system. The origin is at the location of the navigation system, and the three coordinate axes point to the North East High (NEH) direction. The i system is the IMU coordinate system, and there is an installation deviation between the car body coordinate system, b system and IMU coordinate system. The car body coordinate system is recorded as the b system, and the three axes of the car body coordinate system point to the front left and upper direction of the trolley, respectively. In Equation (2), $C_{ib}^R = C_b^R C_i^b$, and C_i^b is the direction cosine matrix from the car system to the IMU coordinate system, representing the installation relationship between the IMU and car body. C_b^R is the direction cosine matrix between the IMU and the navigation system, and the relationship between it and the attitude quaternion is

$$C_b^R = \begin{bmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2(q_1q_2 - q_0q_3) & 2(q_1q_3 + q_0q_2) \\ 2(q_1q_2 + q_0q_3) & q_0^2 - q_1^2 + q_2^2 - q_3^2 & 2(q_2q_3 - q_0q_1) \\ 2(q_1q_3 - q_0q_2) & 2(q_2q_3 + q_0q_1) & q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{bmatrix} \quad (3)$$

which equals

$$C_b^R = \begin{bmatrix} \cos\gamma\cos\varphi + \sin\gamma\sin\varphi\sin\theta & \sin\varphi\cos\theta & \sin\gamma\cos\varphi - \cos\gamma\sin\varphi\sin\theta \\ -\cos\gamma\sin\varphi + \sin\gamma\cos\varphi\sin\theta & \cos\varphi\cos\theta & -\sin\gamma\sin\varphi - \cos\gamma\cos\varphi\sin\theta \\ -\sin\gamma\cos\varphi & \sin\theta & \cos\gamma\cos\theta \end{bmatrix} \quad (4)$$

where φ, θ, γ represent three Euler angles, the heading, pitch and roll angles, respectively.

The trajectory estimation method was based on the combination of the uses of the IMU and odometer. The attitude information measured by IMU and the speed information measured by odometer for trajectory estimation and its position update equation can be expressed as

$$\Delta\tilde{r}^R = C_b^R C_i^b v^b \quad (5)$$

where $\Delta\tilde{r}^R = [\Delta r_N \ \Delta r_E \ \Delta r_H]^T$ is the relative displacement vector of the track measurement car relative to the last observation position under the navigation system. The speed of the vector car was $v^b = [\tilde{v}_0 \ 0 \ 0]^T$ under the car system, and \tilde{v}_0 is the forward speed of the car. When the installation error angle is small, the matrix C_i^b can be expressed as

$$C_i^b = \begin{bmatrix} 1 & \varepsilon_z & -\varepsilon_y \\ -\varepsilon_z & 1 & \varepsilon_x \\ \varepsilon_y & -\varepsilon_x & 1 \end{bmatrix} \quad (6)$$

where $\varepsilon_x, \varepsilon_y, \varepsilon_z$ is the axial installation error of the IMU relative to the X-axis, Y-axis and Z-axis of the trolley. When the odometer had a measurement error, the true speed \tilde{v}_0 can be expressed as

$$\tilde{v}_0 = (1 + \delta k)v_0 \quad (7)$$

where δk is the scale factor error of the odometer, v_0 is the output speed of the odometer. In trajectory measurement, the product of inertial navigation attitude angle error and odometer error δk , and IMU installation error $\varepsilon_x, \varepsilon_y, \varepsilon_z$ is a high-order small quantity and can be ignored.

2.2. Error Equation

In the trajectory estimation, the source of position errors mainly included the initial IMU attitude angle, drift, inertial navigation installation and odometer speed measurement errors. The IMU attitude angle error is obtained from the attitude error equation. The attitude error equation of the system is expressed, as in [22]:

$$\dot{\Phi}^R = -\omega_{iR}^R \times \Phi^R + \delta\omega_{iR}^R - C_b^R \delta\omega_{ib}^b \quad (8)$$

where $\Phi^R = [\varnothing_N \ \varnothing_E \ \varnothing_H]^T$ is the attitude error angle relative to the three axes of the navigation system, $\delta\omega_{iR}^R$ is the angular velocity error vector of the rotation of the navigation system relative to the inertial system; ω_{ib}^b represents the drift error of the three axial gyroscopes; \times denotes a vector cross product.

Since the radius of curvature of the track is usually large and the slope of the track is small, the horizontal attitude angle of the track measurement trolley is usually small, so each element of the direction cosine matrix C_b^R is approximately expressed as

$$C_b^R \approx \begin{bmatrix} \cos\varphi & -\sin\varphi & \theta\cos\varphi + \gamma\sin\varphi \\ \sin\varphi & \cos\varphi & -\gamma\cos\varphi + \theta\sin\varphi \\ -\theta & \gamma & 1 \end{bmatrix} \quad (9)$$

By rotating the angle around the vertical direction, the two horizontal error angles $\varnothing_N, \varnothing_E$ are projected onto the forward and side directions of the trolley. The attitude error

angles in the two directions are represented by $\varnothing_\gamma, \varnothing_\theta$ and the heading error is represented by \varnothing_φ . Its conversion relationship is expressed as

$$\begin{bmatrix} \varnothing_\gamma \\ \varnothing_\theta \\ \varnothing_\varphi \end{bmatrix} = \begin{bmatrix} \cos\varphi & \sin\varphi & 0 \\ -\sin\varphi & \cos\varphi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \varnothing_N \\ \varnothing_E \\ \varnothing_H \end{bmatrix} \tag{10}$$

Substituting Equation (8) into Equations (6) and (7) results in

$$\begin{bmatrix} \dot{\varnothing}_\gamma \\ \dot{\varnothing}_\theta \\ \dot{\varnothing}_\varphi \end{bmatrix} = - \begin{bmatrix} \delta\bar{w}_x^b \\ \delta\bar{w}_y^b \\ \delta\bar{w}_z^b \end{bmatrix} + \begin{bmatrix} \omega_\gamma \\ \omega_\theta \\ \omega_\varphi \end{bmatrix} \tag{11}$$

where $[\delta\bar{w}_x^b \ \delta\bar{w}_y^b \ \delta\bar{w}_z^b]^T$ is the equivalent gyroscope zero bias, and $[\omega_\gamma \ \omega_\theta \ \omega_\varphi]^T$ is the equivalent gyro angle random walk noise.

Due to the weak mobility of the measuring trolley in the railway track surveying, there was a coupling between the systematic errors. There were few observation control points, so it was impossible to effectively estimate all systematic errors. In short-distance measurements, the roll angle of attitude errors did not cause significant position errors in three directions, and the roll angle error could be ignored when constructing the Kalman filter model. Equation (9) is

$$\begin{bmatrix} \dot{\phi}_\theta \\ \dot{\phi}_\varphi \end{bmatrix} = - \begin{bmatrix} \bar{w}_y^b \\ \bar{w}_z^b \end{bmatrix} + \begin{bmatrix} w_\theta \\ w_\varphi \end{bmatrix} \tag{12}$$

Based on the above analysis, the error term was put into Equation (5) to obtain the estimated position value Δr^R with error. According to the position update equation, the obtained position error equation is

$$\delta\Delta r^R = \Delta\tilde{r}^R - \Delta r^R \approx \begin{bmatrix} 0 & \sin\varphi & \cos\varphi \\ 0 & -\cos\varphi & \sin\varphi \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} \varnothing_\theta \\ \varnothing_\varphi \\ \delta k \end{bmatrix} v_0 \tag{13}$$

2.3. RTS Smoothing Algorithm

Since the position of the control point measured by the total station and the estimated trajectory error of the original data were used as the observation update for the filter in the generation of the 3D point cloud based on multi-sensor fusion, when there was an observation update, the error of the measured position estimate and its covariance are both very small. However, the position measurement estimated errors and their covariances between the two observation updates increased over time owing to residual systematic errors. To obtain a high-precision position estimation in the whole process, as is necessary to introduce an appropriate estimation method in the measurement interval without an observation update so as to fully use the observation information before and after the observation time point. This was a typical fixed interval smoothing problem.

The commonly used methods to address the issues of fixed-interval smoothing include two-filter and RTS smoothing. Although both algorithms are theoretically equivalent, the RTS smoothing algorithm was relatively simple to implement. The two-filter smoothing algorithm included two independent Kalman filtering processes, which made it more complicated to implement and required more variables to be stored. The RTS smoothing algorithm combined the reverse filter and the smoothing process into a single reverse recursive form, including a forward Kalman filtering process and reverse smoothing process. The structure was relatively simple, the implementation was convenient, and the calculation efficiency was relatively high. The RTS smoothing algorithm, jointly proposed by Rauch, Tung and Striebel, was a classic fixed interval filtering smoothing algorithm. In this study, we used the RTS smoothing algorithm to process the track measurement interval data.

The RTS smoother consists of a typical forward Kalman filter and a reverse smoother. After each system propagation and observation update, the system state vector, error covariance matrix and state transition matrix were recorded. After the data reached the end, the data were smoothed in reverse from the end to the starting point. The calculation flow of the RTS smoothing algorithm is shown in Figure 1.

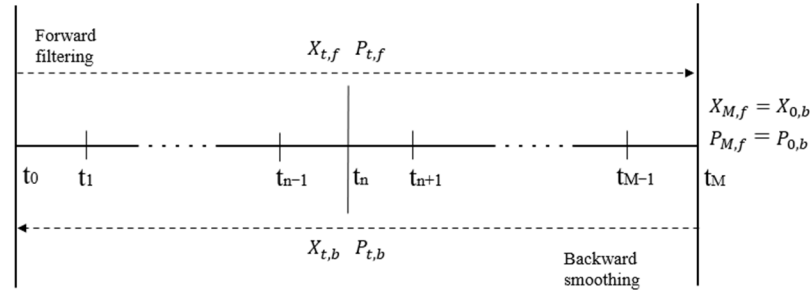


Figure 1. RTS smoothing algorithm flow.

The forward filtering of the RTS smoothing algorithm is a typical Kalman filter, and the smooth estimated value of the state vector was a weighted combination of two filtering estimated values. The calculation is as follows:

$$\begin{aligned}
 X_{t,f}^- &= F_t X_{t-1,f} \\
 P_{t,f}^- &= F_t P_{t-1,f} F_t^T + \Gamma_t Q_t \Gamma_t^T \\
 K_{t,f} &= P_{t,f}^- H_t^T [H_t P_{t,f}^- H_t^T + R_t]^{-1} \\
 X_{t,f} &= X_{t,f}^- + K_t [Z_t - H_t X_{t,f}^-] \\
 P_{t,f} &= [I - K_t H_t] P_{t,f}^-
 \end{aligned} \tag{14}$$

where $X_{t,f}$ and $P_{t,f}$ represent the optimal filter estimate and its covariance matrix of the forward filter state vector at time t ; $X_{t,f}^-$ and $P_{t,f}^-$ represent the optimal one-step prediction value of the forward filter state vector at time t and its covariance matrix. H_t is the observation matrix at time t ; $K_{t,f}$ is the optimal gain matrix of forward filtering at time t ; F_t represents the system state transition matrix and can be obtained from the systematic error matrix A_t . Γ_t is the discrete noise matrix at time t for forward filtering.

The inverse smoothing process can be expressed as

$$\begin{aligned}
 K_{t,b} &= P_{t,f} F_t^T (P_{t+1,f}^-)^{-1} \\
 X_{t,b} &= X_{t,f} + K_{t,b} [X_{t+1,b} - X_{t+1,f}^-] \\
 P_{t,b} &= P_{t,f} + K_{t,b} [P_{t+1,b} - P_{t+1,f}^-] K_{t,b}^T
 \end{aligned} \tag{15}$$

where $X_{t,b}$ and $P_{t,b}$ are the optimal smoothed estimates at time t of the RTS smoothing algorithm and its error covariance matrix, $K_{t,b}$ is the optimal smoothing gain matrix of the RTS smoother at time t .

3. A Method of 3D Point Cloud Generation by Multi-Sensor Fusion

To solve the coordinate calculation of 3D point cloud in non-GNSS scenarios, we propose a 3D point cloud generation method based on multi-sensor fusion. In the track mobile laser measurement system (TMLs), to solve the calibration of the relative position relationship of the scanner, IMU and carrier space, we defined a coordinate system fixed on the sensor integrated frame, which was called the car body coordinate system. Firstly, the method calculated the displacement vector from the control point to the center of the car body according to the calibration parameters using the cloud coordinates of the control

point in the car body coordinate system and the attitude angle of the inertial navigation system at the same time. Next was the calculation of the coordinates of the center point of the same section track according to the coordinates of the control point and the displacement vector. When the track inspection trolley was placed on the track, the center point of the track coincided with the center of the car body. We used the original attitude angle and speed data collected by the IMU and odometer to calculate the center trajectory coordinates of the car body and compare the track center point calculated by the control point with the error of the calculated trajectory point at the same time as the observation update in the Kalman filter. The odometer speed and IMU pitch were corrected using the optimal estimates obtained after filtering and smoothing.

After the attitude error was corrected, the displacement parameters from the control point to the track center point of the same section were changed. The corrected attitude angle was used to recalculate the displacement vector and track center, and the calculated error was compared with the original data to perform secondary filtering. Figure 2 shows the overall algorithm flow chart.

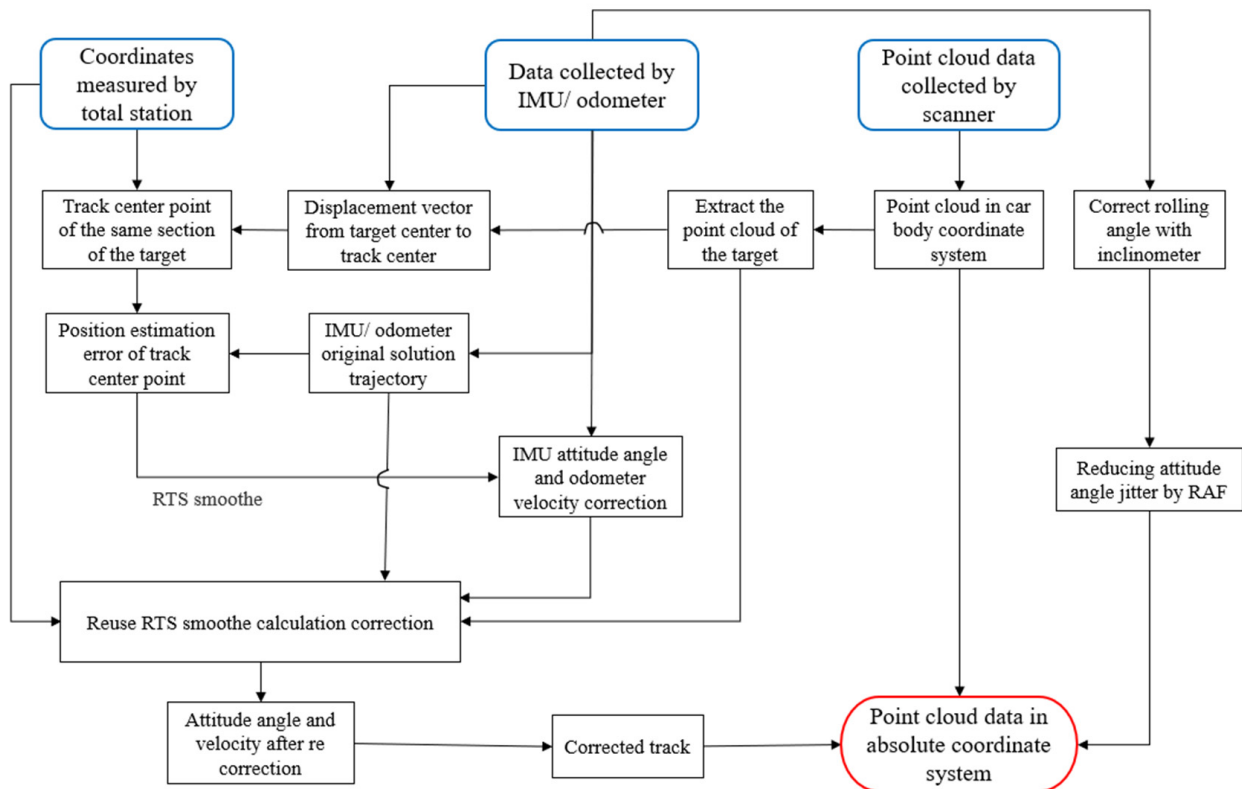


Figure 2. Flow chart of point cloud generation based on RTS filtering and smoothing.

3.1. Research on Track Mobile Laser Measurement System (TMLS)

We mainly integrated sensors such as a Z + F 5016 3D laser scanner, FSINS3X inertial navigation system, inclinometer, rail car and laptop computer in the TMLS, as shown in Figure 3. The track trolley contained a motor, odometer and gauge measuring instruments. It drove at a constant speed on the track at three different speeds of 0.5 m/s, 1.0 m/s, and 1.5 m/s. The odometer scale error was 0.45%. The scanner, IMU, inclinometer, etc. were all controlled by the laptop to start and stop. The IMU, inclination, mileage and gauge data were transmitted to the computer in real time through the serial port and the scanner data was recorded on its own solid-state hard disk.

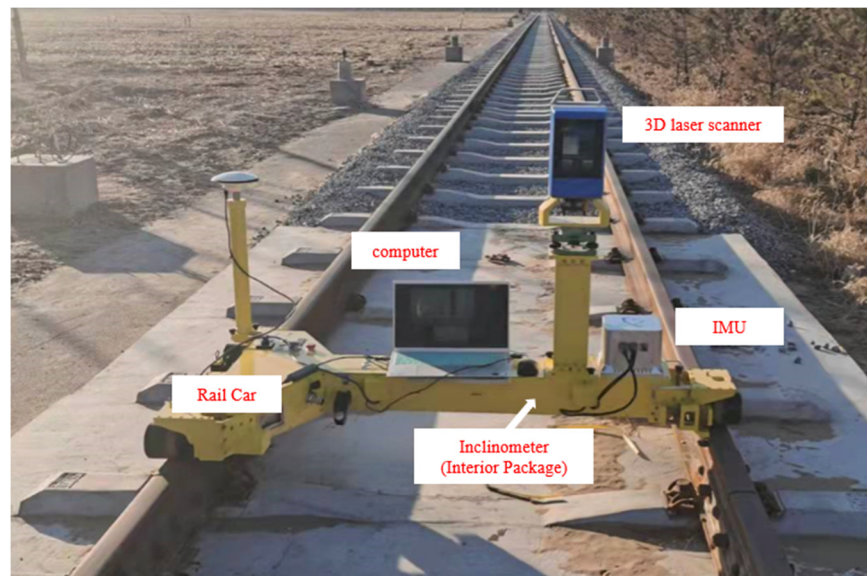


Figure 3. Track Mobile Laser Measurement System (TMLS).

System synchronization mainly included time synchronization and space synchronization. To ensure that the scanner and IMU in the system were consistent in the time system, and that the scanner, IMU and rail car were consistent in the space coordinate system, it was necessary to unify the time and space coordinates of each sensor in the system [23]. Figure 4 presents the coordinate definition of the main sensors in TMLS. The IMU defined the three-dimensional coordinate axis according to the equipment design framework. One X-axis and one Y-axis pointed to the front of the system and the others pointed to the left side of the system. On the right side, the Z-axis was upward. The origin of the scanner coordinate system was the laser emission point, the X-axis was perpendicular to the scanning plane and pointed to the front of the scanner, that is, the front of the scanning section. The Z-axis pointed vertically upward in the scanning plane, and the Y-axis was perpendicular to the scanning plane. The axes were also in the scan plane and perpendicular to the XOZ plane, forming a left-handed coordinate system.

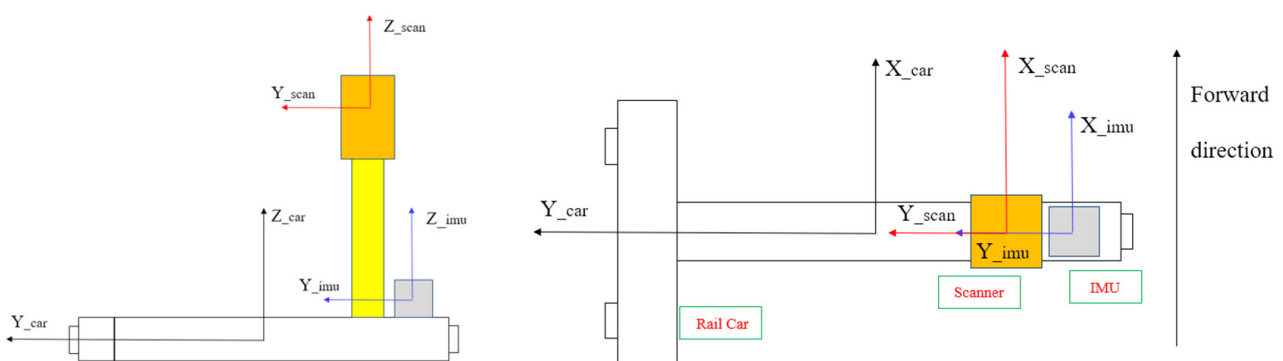


Figure 4. Schematic diagram of the coordinates of the main sensors of TMLS.

In terms of time synchronization, the system recorded the system time uniformly in the IMU, inclination and trolley data using the laptop. A Z + F 5016 scanner can automatically record the start time from the scan start time. The system time of the scanner was time unified. In the space synchronization, we adopted the method of system calibration to unify the space coordinate system.

3.1.1. Hardware Equipment

The TMLS mainly included sensors such as 3D laser scanner, IMU, inclinometer and odometer. The specific models and parameters are as follows.

(1) 3D Laser Scanner

The 3D laser scanner is a sensor that uses the principle of laser ranging to efficiently obtain information such as the 3D coordinates and reflectivity of the measured object. We used the Z + F 5016 three-dimensional laser scanner produced by the Z + F Company in Germany, as shown in Figure 5, and the main technical parameters are shown in Table 1.



Figure 5. Z + F 5016 scanner.

Table 1. Main technical parameters of the s Z + F 5016canner.

Parameter	Standard
Angle measurement accuracy	14.4"
Distance measuring error	$\leq 1\text{mm} \pm 10 \text{ ppm}$ within range
Vertical view	320°
Horizontal view	360°
Scan range	0.3–365 m
Maximum point rate	1,100,000/s
Rotating speed	Maximum 55 Hz
Laser divergence	0.3 mrad
Reflection system	Fully enclosed rotating lens, built-in HDR camera and LED flash

(2) Inertial measurement unit (IMU)

An IMU is a device used to measure the attitude angle and acceleration of an object. We used a FSINS3X optical fiber strapdown inertial navigation system produced by the Harbin Hangshi Enterprise Group, shown in Figure 6. A FSINS3X consists of fiber optic gyroscopes, quartz flexible accelerometers, navigation computers, power modules, etc. It can quickly measure the attitude, speed and position of moving carriers (cars, ships, aircraft, etc.).

Through corresponding settings, the system can be combined with a GNSS, odometer (ODO), Doppler log (DVL) and ultra-short baseline (USBL) to obtain more accurate navigation information. Figure 6 shows the FSINS3X, and Table 2 shows its main technical parameters.



Figure 6. FSINS3X.

Table 2. Main technical parameters of a FSINS3X inertial navigation system.

Index	Parameter	Remark	
Initial alignment	Horizontal attitude	0.02° 0.05° (max)	horizontal attitude angle < 80°
	Course	0.06° 0.10°	static start < 5 min dynamic start < 10 min
Navigation	attitude And Course hold	0.01° /h	pure inertial navigation
Positioning accuracy		0.8 nm/h 0.2%L (L is voyage)	pure inertial navigation combined odometer
Maximum frequency of data transmission		2000 Hz	
Other indicators	Operating temperature	−40–65 °C	
	Input voltage	18~36VDC	
	Input current	<0.6 A	
	Power consumption	<14 W	
	Electrical interface	RS422/CAN/NET	
	Dimensions	166 mm × 151 mm × 126 mm	
	Weight	<4 kg	

(3) Inclination Sensor

The inclination sensor mainly measures the track pitch angle and lateral inclination angle in the stationary phase and is used to assist in correcting the horizontal attitude angle of the IMU. This system selects a BW-VG527 dual-axis inclination sensor (as shown in Figure 7). The weighing accuracy is 0.01°, and the RS232 communication mode was adopted. The performance indicators are shown in Table 3.

3.1.2. System Software Design

We developed a program related to the processing of point cloud data in tunnel scanning based on TMLS using the Visual Studio 2017 platform and C++ language, which mainly included integrated system control, data acquisition, data post-processing and other modules. The diagram of the program design structure is shown in Figure 8.



Figure 7. BW-VG527 Inclinometer.

Table 3. Main technical parameters of a BW-VG527 inclinometer.

Index	Parameter	
Attitude parameters	Dynamic accuracy	0.1°
	Static accuracy	0.01°
	Resolution	0.01°
	Tilt range	±90°
Other indicators	Maximum output frequency	100 Hz
	Voltage	9–35 V DC
	Working current	30 mA
	Operating temperature	−40–85 °C
	Electrical interface	RS232
	Weight	<360 g

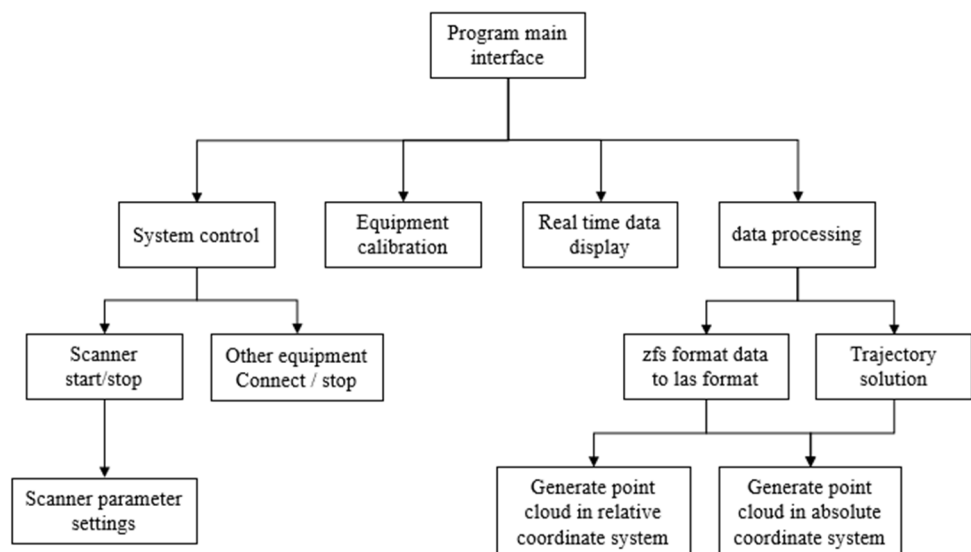


Figure 8. Program design structure diagram.

The program can decode the original file and generate the point cloud data in the Las format, generate trajectory data and generate three-dimensional point cloud coordinates. The main interface is shown in Figure 9. The main functions include system control, data collection, ZFS format conversion to LAS format, trajectory data solution, generation of point cloud data in relative coordinate system and point cloud data in an absolute coordinate system.

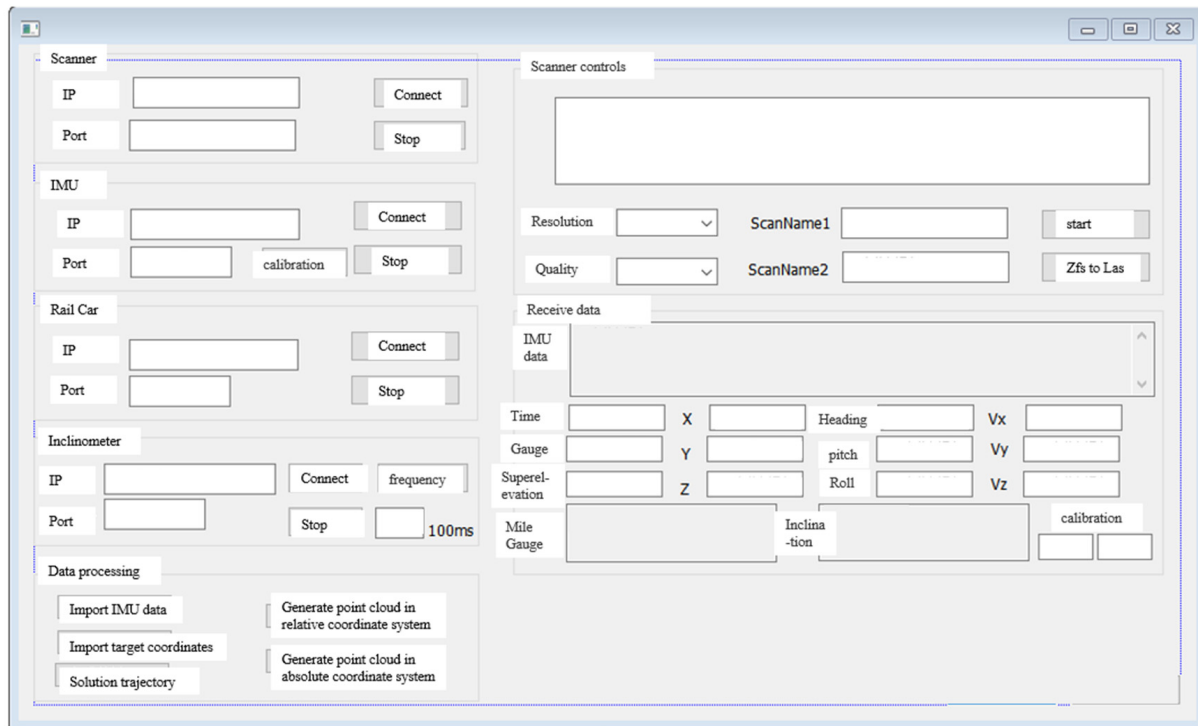


Figure 9. Program interface.

3.1.3. System Calibration

To complete the spatial synchronization of each TMLS sensor, we used a method of system calibration to unify the spatial coordinate system. By calibrating the scanner, the relative position between the scanner center and the car body was obtained, and the scanner coordinate system and the car body coordinate system were unified. By calibrating the IMU, the attitude angle error between IMU and the car body in the engineering coordinate system was obtained, and the IMU coordinate system and the car body coordinate system were unified. To prevent the horizontal attitude angle of IMU and zero-degree value of angle measurement of the inclinometer from being parallel to the horizontal plane and closure error of the IMU heading angle [24], the zero-degree deviation of horizontal inclination and the closure error of the heading angle were calibrated.

(1) Scanner Calibration

In the scanner calibration, the total station and target points were used to calculate the conversion parameters between the scanner coordinate system and the car body coordinate system. First, we stuck four reflectors on the tunnel wall at the same section and stuck four reflectors on the rail car. We used the total station to measure the coordinates of the centers of these seven reflectors. The scanner was turned on to scan the area where the target was located and extracted the target center coordinates from the point cloud, as shown in Figure 10.

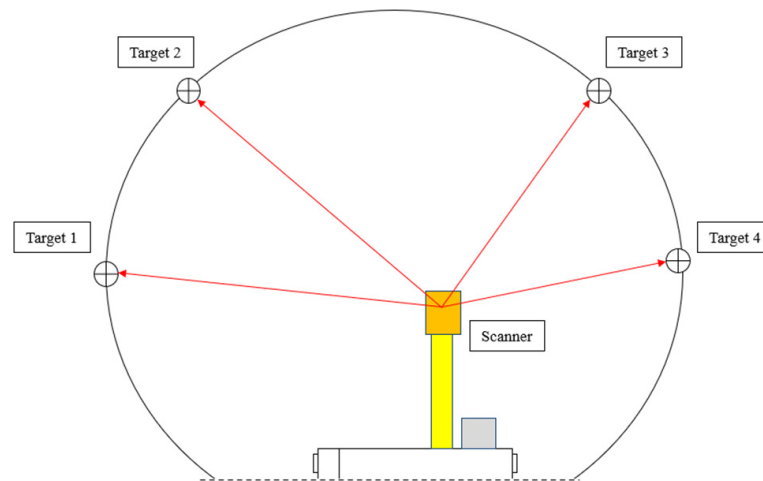


Figure 10. Scanner Calibration.

To calculate the calibration parameter, we first defined the car body coordinate system (RC), the scanner coordinate system (RS) and the total station coordinate system (RT). First, the conversion parameters between RT and RS were calculated through the center points of the four reflectors on the tunnel wall. Then we calculated the conversion parameters between RT and RC through the three common points on the car body. Finally, the conversion parameters between RS and RC were obtained through two translation rotations, and then the point cloud data could be converted from RS to RC.

(2) IMU Calibration

The IMU coordinate system (RI) and RC could not be completely coincident due to the influence of the instrument installation error; therefore, it was necessary to calibrate the IMU installation angle error. Figure 11 shows the sensor coordinate systems and their relative relations.

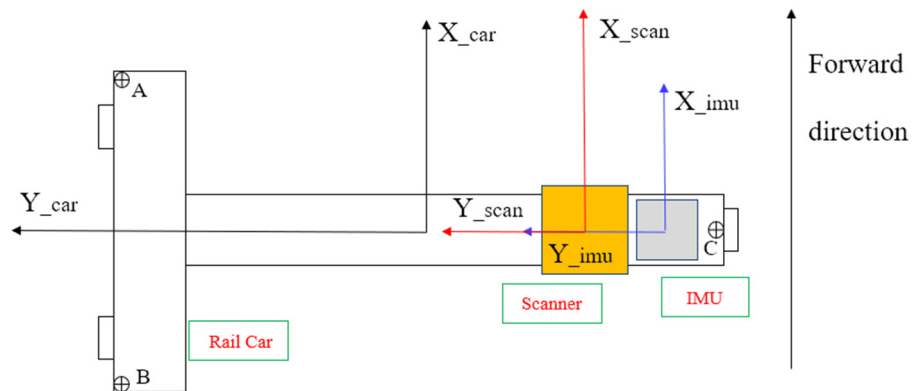


Figure 11. Relationship between the plane coordinate systems of each sensor.

In the IMU calibration, the total station was used to measure the engineering coordinate system coordinates of the three points A, B and C, on the body of the rail car. According to Equation (16), the attitude angle of the car at that time could be calculated and then compared with the attitude angle observed by the IMU; the difference of the three-axis attitude angle between RI and RC could be obtained. The average value was obtained through multiple measurements of the error calibration of the IMU installation angle as the result.

$$\begin{aligned}
 \text{Heading} &= \arctan\left(\frac{y_A - y_B}{x_A - x_B}\right) \\
 \text{Pitch} &= \arcsin\left(\frac{z_A - z_B}{\sqrt{(x_A - x_B)^2 + (y_A - y_B)^2 + (z_A - z_B)^2}}\right) \\
 \text{Roll} &= \arcsin\left(\frac{z_C - (z_A + z_B)/2}{\sqrt{(x_C - (x_A + x_B)/2)^2 + (y_C - (y_A + y_B)/2)^2 + (z_C - (z_A + z_B)/2)^2}}\right)
 \end{aligned} \tag{16}$$

(3) Horizontal inclination zero deviation and heading angle closure difference calibration

We used a method of positive and negative measurement for calibration. Before operation, we placed the stationary trolley on the track facing the direction of measurement to measure the angle, then turned the trolley 180° and measured again. As shown in Figure 12, the error between the measured horizontal inclination angle and the actual inclination angle was $\Delta\alpha$. The actual inclination angle as β ; therefore, the measured value was $A_1 = \beta + \Delta\alpha$. The correct inclination $\beta = \frac{(A_1 - A_2)}{2}$. The inclination correction number between the inclinometer and the track coordinate system could be obtained using $\varepsilon_1 = -\Delta\alpha = -\frac{(A_2 + A_1)}{2}$. The forward measurement value of IMU heading angle was φ_f and the reverse measurement value was φ_b . The sum of the round-trip measured values of IMU direction angle should be 360°. The heading closure difference was $\Delta\varphi = 360 - (\varphi_f + \varphi_b)$ and the heading angle correction number was $\varepsilon_2 = -\Delta\varphi$.

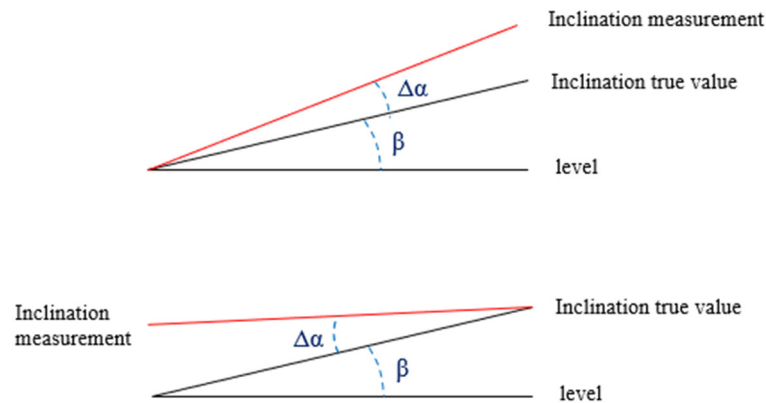


Figure 12. Schematic diagram of inclination calibration.

3.2. Trajectory Correction Algorithm

During the trajectory correction of the scanning trolley, first the coordinates of the track center point of the same section were calculated according to the target point cloud coordinates and IMU attitude angle in RC. The IMU attitude angle and odometer speed were used to calculate the track center coordinates of the next point from the previous point and compare the calculated coordinates of the target control point to obtain the estimated error as an observation update to correct the odometer speed and IMU attitude angle. Since the calculated track center point changed after correcting the attitude angle, the track center point was recalculated using the corrected attitude angle and compared with the original target to determine whether the error exceeds the threshold. If the error exceeded the threshold, a secondary filtering was performed to correct it, but it did not exceed the calculated trajectory. Figure 13 shows the trajectory correction process.

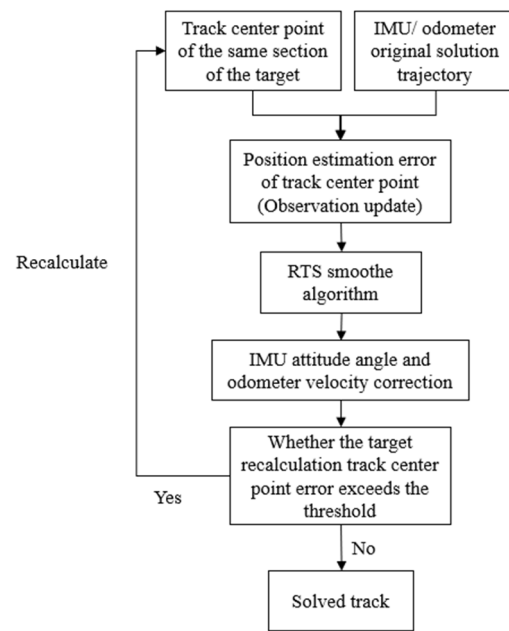


Figure 13. Trajectory correction.

(1) Track center point calculation

After extracting the target point cloud, according to the IMU attitude angle and car body parameters, we calculated the displacement vector ($\Delta X, \Delta Y$ and ΔZ) of each target and the center point of the track after the conversion.

The body correction in RS was:

$$dh = H_{car} \cos \gamma + L_{car} \sin \gamma dl = H_{car} \sin \gamma - L_{car} \cos \gamma \tag{17}$$

where H_{car} is the vertical translation of the scanner center relative to the car body center, and L_{car} is the lateral translation of the scanner center relative to the car body center, and γ is the roll angle at this moment.

The point cloud coordinates in the car body coordinate system were:

$$\begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix} = \begin{bmatrix} x_0 \\ y_0 + dl \\ z_0 + dh \end{bmatrix} \tag{18}$$

According to time alignment, we calculated the rotation parameters of each scanning section according to the IMU attitude angle and then obtained the displacement vector of the target relative to the track center at the same section, as shown in Figure 14:

$$\begin{bmatrix} x_{c_t} \\ y_{c_t} \\ z_{c_t} \end{bmatrix} = R_X R_Y R_Z \begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix} \tag{19}$$

where R_X, R_Y, R_Z are calculated by the following equations:

$$R_X = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \varphi & \sin \varphi \\ 0 & -\sin \varphi & \cos \varphi \end{bmatrix} R_Y = \begin{bmatrix} \cos \theta & 0 & -\sin \theta \\ 0 & 1 & 0 \\ \sin \theta & 0 & \cos \theta \end{bmatrix} R_Z = \begin{bmatrix} \cos \gamma & \sin \gamma & 0 \\ -\sin \gamma & \cos \gamma & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{20}$$

where φ, θ, γ represent three Euler angles, which are heading, pitch and roll angles.

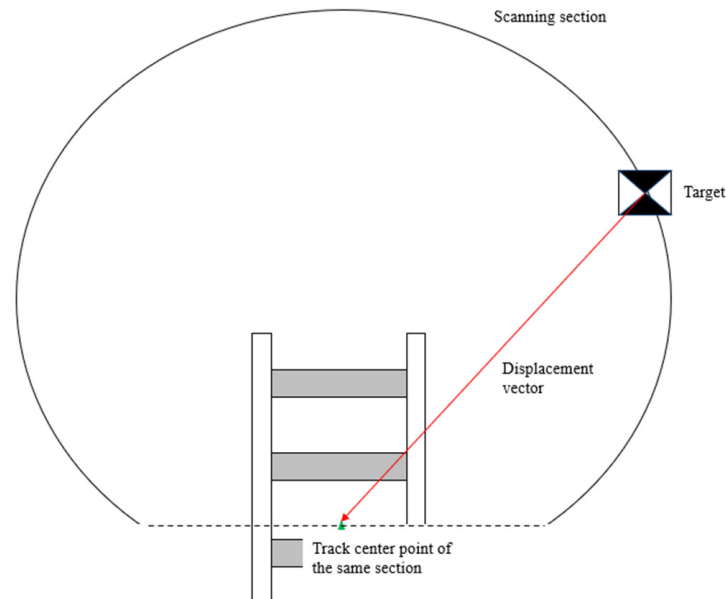


Figure 14. Displacement vector from the center of the target to the center of the track in the same section.

According to the position of the control point, the track center point of the section sta_{abs} could be inversely calculated. This was used as a control point to compare the error with the trajectory calculated from the original acquisition data of the IMU and odometer.

$$sta_{abs} = \begin{bmatrix} x_{tag} \\ y_{tag} \\ z_{tag} \end{bmatrix} - \begin{bmatrix} x_{c_t} \\ y_{c_t} \\ z_{c_t} \end{bmatrix} \tag{21}$$

where $(x_{tag}, y_{tag}, z_{tag})$ is the coordinate of the target center measured by the total station.

(2) Original data estimation error

According to the position update algorithm, the IMU attitude angle and odometer speed were used to calculate the line trajectory, and according to the position update equation, the approximate displacement increment ΔS of the time $t + \Delta t$ relative to the time t was calculated, and the time $t + \Delta t$ coordinates were

$$r_{t+\Delta t} = r_t + \Delta S \tag{22}$$

We obtained the track center point at the same time as the target center point, and translated the original data to estimate the trajectory, so that the previous target calculated track center point coincided with the corresponding time estimated trajectory point, and the error $[\delta\Delta r_N \ \delta\Delta r_E \ \delta\Delta r_H]^T$ between the trajectory coordinates and the target calculated track center point at the next moment could be calculated.

(3) RTS filter correction

In the system, the IMU zero bias instability was 0.01/h, the angle random walk noise was 0.02/h, the odometer measurement accuracy was 0.45/1000, the target center point coordinate measurement was measured by Leica high-precision optical total station, the angle measurement accuracy was 0.5", and the ranging accuracy was $1 \text{ mm} + 2 \times 10^{-6}$.

The trajectory estimation error of the control point corresponding to the track center was used as the observation update and the optimal estimation of the IMU attitude angle error and the odometer velocity error was obtained by filtering, which was used to correct the original data and solve the track center line. The discrete Kalman filter equations in the horizontal and vertical directions were:

$$\begin{aligned}
 X_l(t) &= F_l X_l(t-1) + w_l(t-1) \\
 Z_l(t) &= H_l X_l(t) + \mu_l(t) \\
 X_v(t) &= F_v X_v(t-1) + w_v(t-1) \\
 Z_v(t) &= H_v X_v(t) + \mu_v(t)
 \end{aligned}
 \tag{23}$$

Since there was no coupling between the horizontal and vertical directions, Kalman filter models could be established for the horizontal and vertical directions, respectively. We selected the pitch angle, heading angle attitude error \varnothing_θ and \varnothing_φ , gyro offset sum $\delta\bar{w}_y^b$ and $\delta\bar{w}_z^b$, position error $\delta\Delta r_N, \delta\Delta r_E, \delta\Delta r_H$, and odometer scale factor δk as Kalman filter state quantities. The system state variables in the horizontal and vertical directions were

$$X_l(t) = \begin{bmatrix} \delta\Delta r_N \\ \delta\Delta r_E \\ \delta k \\ \varnothing_\varphi \\ \delta\bar{w}_z^b \end{bmatrix}, \quad X_v(t) = \begin{bmatrix} \delta\Delta r_H \\ \varnothing_\theta \\ \delta\bar{w}_y^b \end{bmatrix}
 \tag{24}$$

The displacement error observation was used as the filter update, the position observation value was provided by the total station, and the difference between the position value and the position value measured by dead reckoning was used as the observation update of the Kalman filter, namely,

$$Z_l(t) = \begin{bmatrix} \delta\Delta r_N \\ \delta\Delta r_E \end{bmatrix}, \quad Z_v(t) = \delta\Delta r_H
 \tag{25}$$

where the system transition matrix F_l is

$$F_l = \begin{bmatrix} 1 & 0 & v_0 t \cos\varphi & v_0 t \sin\varphi & -\frac{1}{2} v_0 t^2 \sin\varphi \\ 0 & 1 & v_0 t \sin\varphi & v_0 t \cos\varphi & \frac{1}{2} v_0 t^2 \cos\varphi \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & -t \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}
 \tag{26}$$

$$F_v = \begin{bmatrix} 1 & v_0 t & \frac{1}{2} v_0 t^2 \\ 0 & 1 & -t \\ 0 & 0 & 1 \end{bmatrix}
 \tag{27}$$

Horizontal system noise $w_l(t) = [w_{\delta k} \ w_\varphi]$, $w_l(t) \sim N(0, Q_l(t))$, vertically $w_v(t) = w_\theta$, $w_v(t) \sim N(0, Q_v(t))$, where $Q_l(t) = \text{diag}\left(\left[\sigma_{w_{\delta k}}^2 \ \sigma_{w_\varphi}^2\right]\right)$, $Q_v(t) = \sigma_{w_\theta}^2$ is the system noise power spectral density.

The observation matrices in the horizontal and vertical directions were

$$H_l = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix}
 \tag{28}$$

$$H_v = [1 \ 0 \ 0]
 \tag{29}$$

The horizontal direction observation noise could be expressed as $\mu_l(t) = [\mu_{r_N} \ \mu_{r_E}]^T$, $\mu_l(t) \sim N(0, R_l(t))$; the vertical direction is $\mu_v(t) = \mu_{r_H}$, where $\mu_{r_H}(t) \sim N(0, R_v(t))$ and $R_l(t) = \text{diag}\left([\sigma_r^2 \ \sigma_r^2]\right)$; $R_v(t) = \sigma_r^2$ is the observed noise power spectral density.

When there are observations, they were updated to get the best estimate:

$$\begin{aligned}
 K_{t,f} &= P_{t,f}^- H_t^T \left[H_t P_{t,f}^- H_t^T + R_t \right]^{-1} \\
 X_{t,f} &= X_{t,f}^- + K_t \left[Z_t - H_t X_{t,f}^- \right] \\
 P_{t,f} &= [I - K_t H_t] P_{t,f}^-
 \end{aligned}
 \tag{30}$$

If there was no observation update time, only the filter estimation was performed:

$$\begin{aligned} X_{t+1,f}^- &= F_t X_{t,f} \\ P_{t+1,f}^- &= F_t P_{t,f} F_t^T + Q_t \end{aligned} \tag{31}$$

Reverse smoothing started at the last moment, and the stated estimate and covariance matrix obtained at the last moment by forward recursion were the initial state estimate. The covariance matrix of the backward recursion process, which were $X_{0,b} = X_{M,f}$ and $P_{0,b} = P_{M,f}$, smoothed backward calculation according to Equation (15).

After the attitude error was corrected, the displacement vector of the target to the center point of the track of the same section changed so that the track center was recalculated using the corrected displacement vector. The calculated error was compared with the original data of the IMU and odometer, and the calculated error was filtered for a second time. The IMU attitude angle and odometer speed were corrected using the error parameters obtained after secondary filtering, and the corrected trajectory data was obtained by calculation.

3.3. Attitude Angle Optimization Algorithm

RTS filtering smoothing only corrected the two attitude angles of heading and pitch and not the roll angle. The roll angle error in the generation of three-dimensional point cloud led to a torsion of point cloud. We used a high-precision inclinometer to constrain the roll angle of the entire inertial navigation system. Due to the high acquisition frequency of inertial navigation data, the attitude angle was prone to periodic high-frequency jitter, resulting in the deformation of the generated three-dimensional point cloud data. For this, the recursive average filtering algorithm was used to optimize the three-axis attitude angle and suppress the periodic noise interference at high acquisition frequency.

(1) Inclinometer correction

The roll angle was first corrected according to the high-precision inclinometer. Since the static measurement accuracy of the inclination angle is high, the error in the dynamic measurement becomes larger; therefore, the IMU roll angle is corrected using the static angle measurement at the start and end of the trolley scan. The IMU and inclinometer roll angles measured in the static state before the trolley scanning starts are recorded as γ_{imua} and γ_{inca} , and the difference between the two is calculated as $\Delta\gamma_a$. After the trolley scanning, the statically measured IMU and inclinometer roll angles are recorded as γ_{imub} and γ_{incb} , and the difference is calculated as $\Delta\gamma_b$. Figure 15 shows the inclinometer roll angle and IMU roll angle.

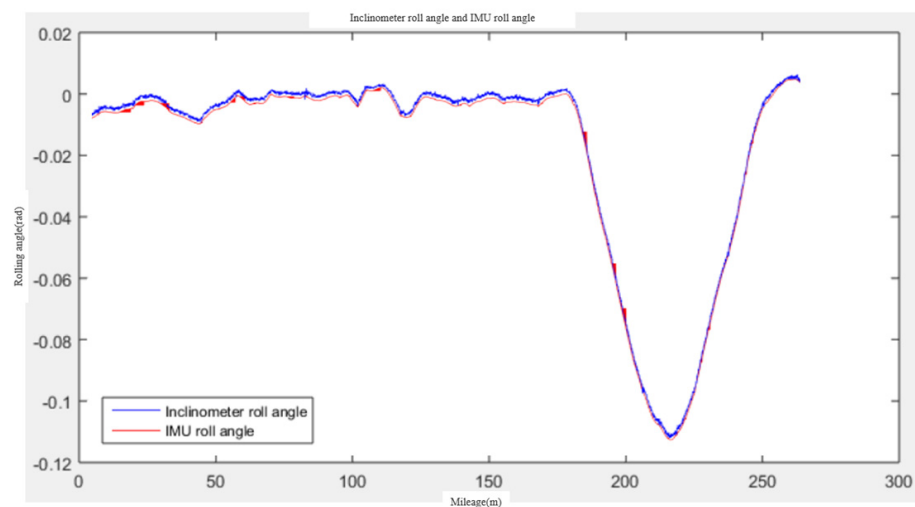


Figure 15. Inclinometer roll angle and IMU roll angle.

First, the roll angle of the IMU was modified according to the inclinometer data, and then the error was evenly distributed according to the scan time. Assuming that the overall scan time was T_{ab} , and the starting time was t_0 , then the IMU data t_i now are $\gamma_{imu}i$, and the roll angle at this moment is

$$\gamma_i = \gamma_{imu}i + \Delta\gamma_a + (t_i - t_0) \frac{\Delta\gamma_b - \Delta\gamma_a}{T_{ab}} \tag{32}$$

(2) Recursive Averaging Filter

Since the attitude angle collected by IMU had the characteristics of a high collection frequency and periodic change, the attitude angle was prone to periodic high-frequency jitter. To avoid impact on the point cloud generation, a method of recursive average filtering was used to optimize the corrected three-axis attitude angle to suppress this high-frequency periodic error interference.

The recursive average filtering method regarded N consecutive sampling values as a queue, and the length of the queue was fixed as N . Each time a new data point was sampled and placed at the end of the queue, the original data at the head of the queue were discarded according to the principle of first-in, first-out. The N data in the queue were arithmetically averaged to obtain a new filtering result. It had a good inhibitory effect on periodic interference, and the smoothness was high, which was suitable for high-frequency oscillation systems. Figure 16 shows the comparison curve before and after recursive average filtering. It was seen from the filtered and the original attitude angles that while maintaining the overall consistency, the method eliminated high-frequency jitter and oscillation, and the data was smooth and more stable.

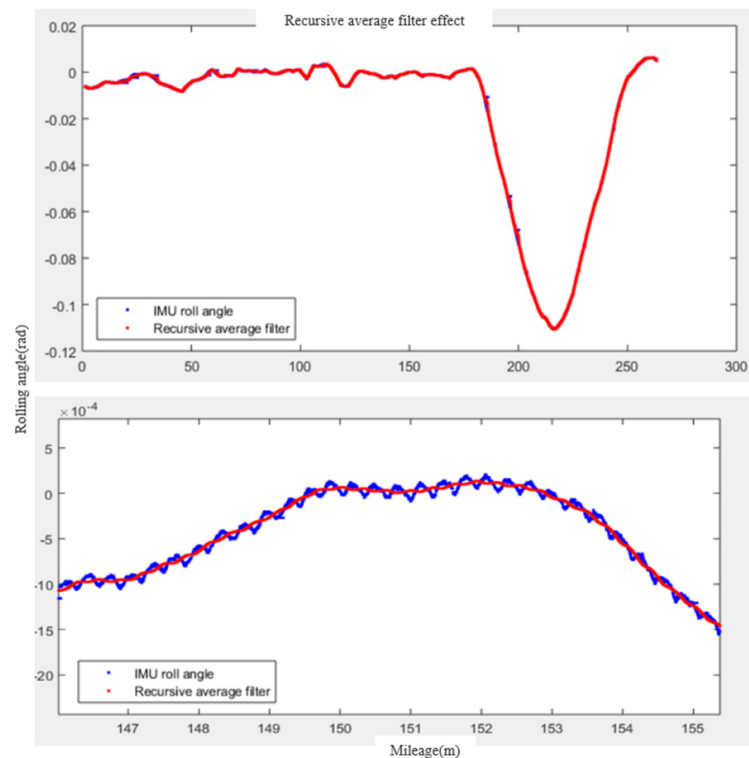


Figure 16. Recursive average filter effect.

3.4. 3D Point Cloud Data Generation Method

According to the transformation relationship between the coordinate systems and the model parameters, the scan data were converted from RS to RC using the calibration parameters of the system. Combined with the optimized attitude angle data and RTS smooth

correction and solution trajectory data, the point cloud data in the absolute coordinate system were obtained by converting the point cloud data in RC.

After filtering and correction, the coordinate of the center point of the track were $[X_c \ Y_c \ Z_c]^T$, the coordinates of the point cloud in RC were $[x_1 \ y_1 \ z_1]^T$, and then the coordinates of the point cloud in the absolute coordinate system were

$$\begin{bmatrix} X_{abs} \\ Y_{abs} \\ Z_{abs} \end{bmatrix} = \begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix} + R_X R_Y R_Z \begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix} \quad (33)$$

Through the point cloud calculation formula of Formula (33), the original data obtained by the scanner were solved into three-dimensional point cloud data in the absolute coordinate system by using the post-processing data of the inertial navigation unit and the transformation principle of the coordinate system. Figure 17 shows the overall effect of the points after the absolute 3D point cloud was generated, in which the tags were the positions of the four target points measured by the total station.

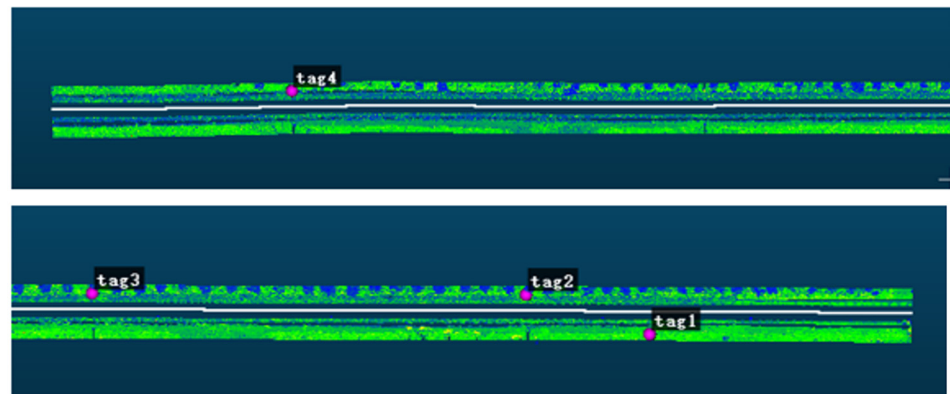


Figure 17. Point cloud generation overall effect.

4. Results

To verify the 3D point cloud generation method based on multi-sensor fusion, a test experiment was conducted at a track test site in Beijing. The test section was about 260 m and included straight and curved sections. As shown in Figure 18, there were multiple pairs of CPIII control point piles beside the experimental track line. In the experiment, a GNSS and IMU integrated navigation system was installed to obtain trajectory data, which were used to compare the deviation with the trajectory data calculated by the RTS smoothing algorithm. We used the integrated navigation method to generate 3D point cloud data and compared it with the multi-sensor fusion method to generate a 3D point cloud to verify the point accuracy and root mean square error.

4.1. Experimental Overview

We verified the experimental data of the test site, hardware system and research algorithm. In the experiment, the car with the integrated navigation system was installed as a reference to obtain the trajectory data and generate the 3D point cloud. These values were compared to the 3D point cloud generation method based on multi-sensor fusion. In the integrated navigation system, the GNSS obtained the satellite observation information, and the IMU obtained the three-axis attitude angle. The GNSS reference station was manually erected to improve positioning accuracy. The GNSS and IMU integrated navigation track measurement system [25] had a lateral deviation accuracy of 2 mm and a vertical deviation measurement accuracy of 2 mm.

This section of the track consisted of straight and curved sections. Four CPIII control pile points with different distances were selected as control points on the line to verify the correction effect and accuracy at different distances. The control points were measured

by a high-precision Leica total station and polar coordinate method. At the same time, satellite receivers were installed at a Leica total station and outside the site to facilitate the orientation of the total station and the conversion between satellite coordinate data and the engineering coordinate system. Figure 19 shows the overall distribution of targets, Figure 20 shows the target photo (left) and the point cloud extraction target (right).



Figure 18. Test site (left) and target photos (right).



Figure 19. Overall distribution of the target.

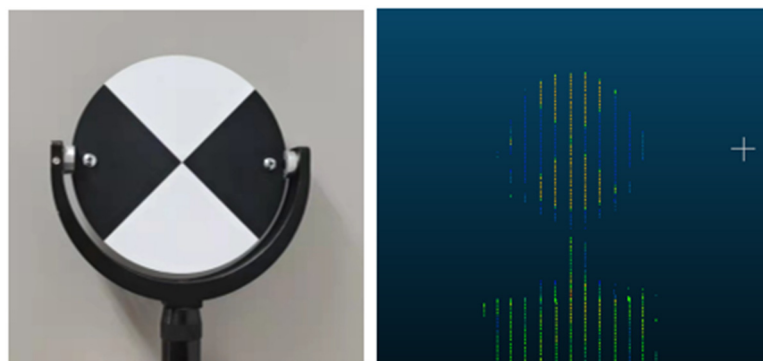


Figure 20. Target photo (left) and Point cloud extraction target (right).

4.2. Filtered Smooth Trajectory Verification

In the experiment, firstly, the trajectory was calculated before and after the correction of the IMU attitude angle, and the odometer speed was compared with the trajectory obtained by integrated navigation. Figure 21 shows the error comparison of the trajectories before and after the filter correction and the integrated navigation solution. Before the filter correction, the trajectory deviation gradually increased with an increase in distance. After the filter correction, the error level was greatly improved. In Figure 21, “a simple IMU position calculation deviations”, the track coordinate was obtained by using only the original data of the IMU and odometer.

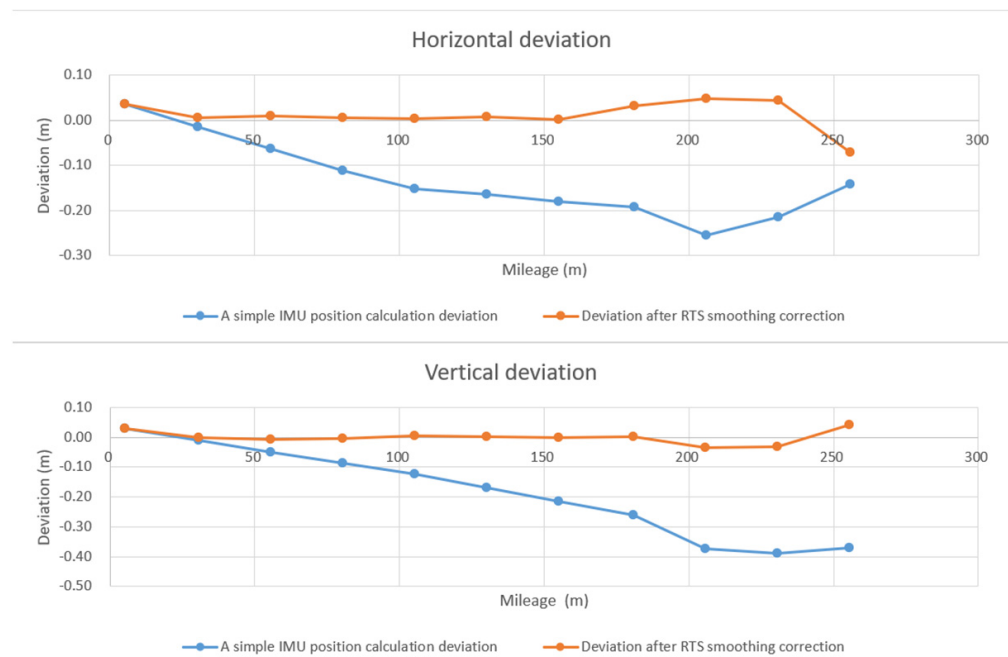


Figure 21. Comparison of trajectory error before and after filter correction.

As shown in Table 4, the trajectory deviation before filter correction gradually increased with the distance. After filtering and correction, the maximum lateral deviation was 0.0482 m, the average deviation was 0.0239 m, the vertical maximum deviation was 0.0405 m, and the average deviation was 0.0138 m.

Table 4. Comparison of trajectory deviation between before and after filter correction and integrated navigation.

Mileage Location(m)	Deviation before Filtering Correction		Deviation after Filtering Correction	
	Lateral Deviation(m)	Vertical Deviation(m)	Lateral Deviation(m)	Vertical Deviation(m)
5	0.0347	0.0307	0.0347	0.0307
30	-0.0145	-0.0089	0.0052	-0.0009
55	-0.0633	-0.0492	0.0086	-0.0056
80	-0.1116	-0.0862	0.0060	-0.0026
105	-0.1523	-0.1229	0.0028	0.0038
130	-0.1644	-0.1685	0.0080	0.0018
155	-0.1794	-0.2135	0.0021	0.0000
180	-0.1916	-0.2592	0.0312	0.0026
205	-0.2538	-0.3729	0.0482	-0.0330
230	-0.2150	-0.3895	0.0440	-0.0301
255	-0.1420	-0.3692	-0.0420	0.0405
Average	0.1384	0.1882	0.0239	0.0138

4.3. Accuracy of Point Cloud Restoration

To verify the accuracy of the point cloud restoration, the restored point cloud target and the control point target were first compared, and then several combined navigation solution point clouds and multi-sensor fusion generated point cloud data were selected for comparison with the same name feature points. Analysis showed that the root mean square error of the feature points with the same name was 0.0214 m. Figure 22 shows the point cloud extraction target and target measurement position. Table 5 is the Point cloud extraction target and target measurement error. Figure 23 shows the multi-sensor fusion generated point cloud and integrated navigation generated point cloud with the same name point selection.

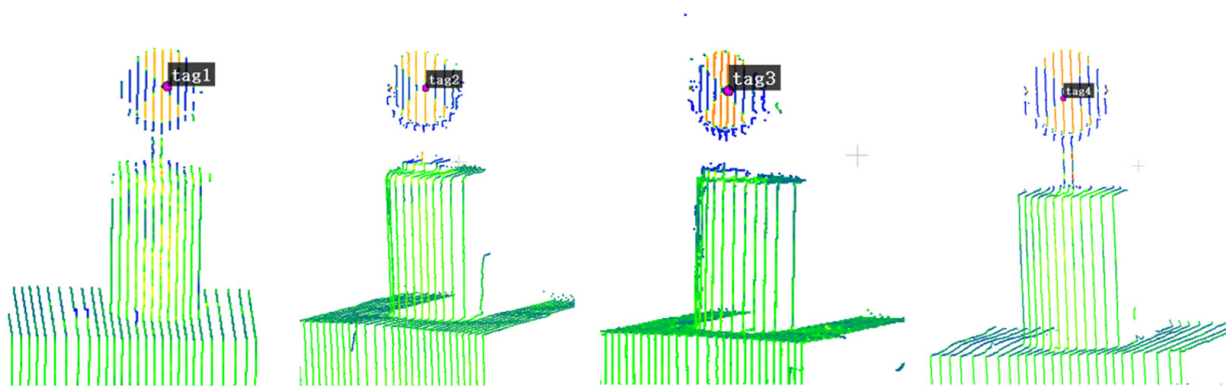


Figure 22. Point cloud extraction target and target measurement position.

Table 5. Point cloud extraction target and target measurement error.

Target	North Error(m)	East Error(m)	Height Error(m)	Horizontal Deviation(m)	Elevation Deviation(m)
tag1	0.0107	0.0000	0.0000	0.0107	0.0000
tag2	−0.0081	0.0008	0.0003	0.0081	0.0003
tag3	−0.0107	0.0038	0.0005	0.0113	0.0005
tag4	0.0276	0.0129	0.0012	0.0305	0.0012
Average	0.0143	0.0044	0.0005	0.0152	0.0005

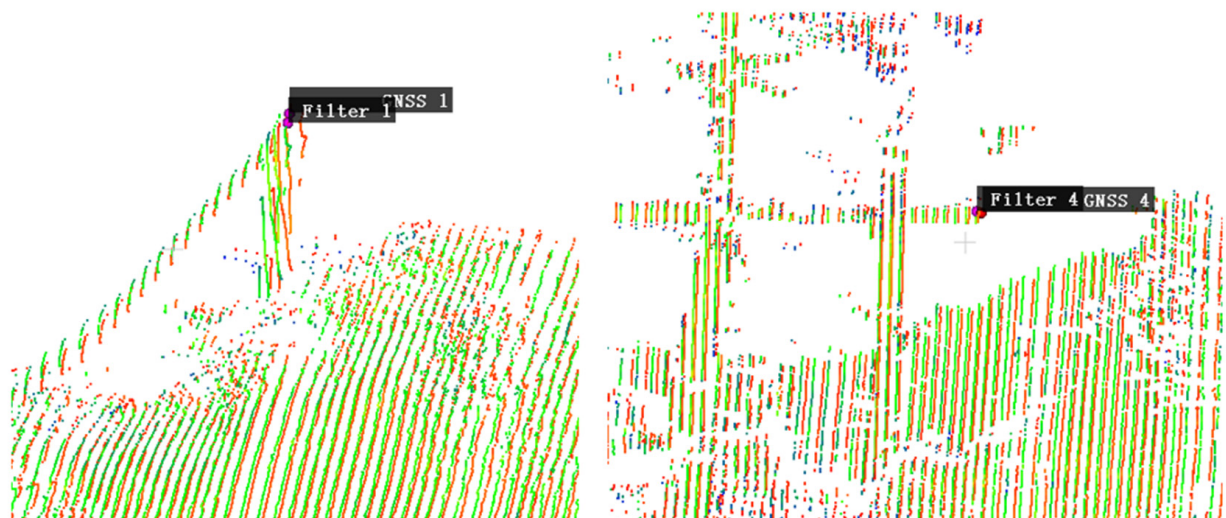


Figure 23. Multi-sensor fusion generated point cloud and integrated navigation generated point cloud with the same name point selection.

As shown in Table 6, comparing the three-dimensional point cloud data generated by multi-sensor fusion and the three-dimensional point cloud data generated by integrated navigation, the overall point error could be controlled within 0.04 m. The maximum deviation of points in the horizontal direction was 0.036 m, and the average deviation was 0.0226 m; the maximum deviation in the elevation direction was 0.0391 m, and the average deviation was 0.0136 m.

Table 6. Error of the same name point generated by multi-sensor fusion and point cloud generated by integrated navigation.

Point ID	North Error(m)	East Error(m)	Height Error(m)	Horizontal Deviation(m)	Elevation Deviation(m)
1	−0.0126	0.0105	0.0009	0.0164	0.0009
2	0.0081	−0.0082	0.0031	0.0115	0.0031
3	−0.0089	0.0294	0.0391	0.0307	0.0391
4	0.0072	0.0294	−0.0054	0.0303	0.0054
5	−0.0073	0.0341	−0.0132	0.0348	0.0132
6	−0.0089	0.0349	−0.0199	0.0360	0.0199
Average	0.0088	0.0244	0.0136	0.0266	0.0136

5. Discussion

The value of the tunnel laser measurement system and 3D point cloud generation method can be discussed from the following aspects.

First, from the working mode, most of the existing railway scanning systems obtain 3D point cloud data based on GNSS and IMU integrated navigation and scanner fusion. Since there is no GNSS signal in the tunnel scene, the traditional tunnel relative scanning system mainly integrates the scanner and odometer to obtain the point cloud data under the relative coordinate system expanded by mileage. To solve the problem of obtaining the absolute coordinates and real geometric forms of point cloud data in a relative coordinate system, Mei et al. and Du et al. both adopted the method of combining design data with point cloud data in a relative coordinate system to generate three-dimensional point cloud data in an absolute coordinate system based on mileage correspondence. However, this method has problems in that construction may not conform to the design. To solve this problem, Han et al. proposed a method of designing a data correction trajectory based on measurement data fitting to generate 3D point cloud data in an absolute coordinate system. Some other teams have adopted another method, directly establishing an absolute scanning system, integrating scanners, inertial navigation and odometers and other auxiliary means to make it suitable for tunnel scenes. A typical case is the Amberg GRP5000. The Amberg GRP5000 integrates a scanner, inertial navigation and odometer; a prism is placed above the scanner. Stopping the trolley at intervals during scanning and using the total station to measure the prism coordinates corrects the trajectory. The TMLS established in this study integrates a scanner, inertial navigation, odometer and inclinometer. Placing targets in the scanning section does not need to stop measuring control points halfway, which can improve efficiency, simplify the operation process and make the generated point cloud data more uniform and stable.

Secondly, from the perspective of measurement accuracy, generally based on the combination of relative scanning and design data, the absolute accuracy of the point cloud is about 0.1 m. The maximum error of the Amberg GRP5000 point cloud is 30.3 mm. The absolute measurement accuracy of the point cloud in this study is 40 mm. In the tunnel scene, the measurement accuracy of this method was slightly lower than Amberg's, but the efficiency was improved.

Finally, at present, we are also studying the point cloud SLAM algorithm. SLAM is more often used for 3D point cloud reconstructions in GNSS-free scenarios. However, due to the characteristics of the tunnel scene with few feature points, the SLAM algorithm is challenged, and corresponding research is needed to adapt it to the tunnel scene.

The method we proposed in this study could effectively avoid the deformation of the tunnel and the sharp increase of the error and provide better data support for follow-up tunnel analyses of the rail transit tunnel, such as three-dimensional displays, completion measurements, disease system management and so on.

This study has achieved certain results in the research of mobile scanning point cloud generation, but there are still problems that need to be further studied. In the system integration, since the PPS pulse signal cannot be used in a non-GNSS environment, the

time synchronization between the scanner and the IMU equipment is synchronized with the computer system time, and there is a risk of time misalignment. We need to continue to explore more accurate time synchronization methods in our follow-up.

6. Conclusions

In areas where there is no GNSS signal, such as in the tunnels, relying solely on IMU calculation results in positioning and attitude errors rapidly accumulating beyond the limit. We proposed TMLS, a mobile laser scanning measurement system with a high-precision laser scanning sensor as the main body and integrated multi-type sensors mounted on rail cars. The method mainly integrated sensors such as a Z + F 5016 3D laser scanner, FSINS3X inertial navigation system, inclinometer, rail car and laptop computer in the TMLS, and developed related processing programs to realize the original point cloud. The decoding of the file generated Las data, the solution generated trajectory data, and the three-dimensional point cloud coordinates were solved. A 3D point cloud generation method based on multi-sensor fusion was designed. The method used the target as the control point, applied the RTS smoothing algorithm to correct the attitude angle and odometer speed of the IMU, calculated the trajectory data, used a high-precision inclinometer to correct the IMU roll angle, and used a recursive average filtering algorithm to optimize the three-axis attitude angle and suppress high-frequency jitter noise. Finally, 3D point cloud data were generated.

Finally, the hardware system and the algorithm were experimentally verified in a test experiment. The 3D point cloud generation method based on multi-sensor fusion was verified in the track scene. The experimental data was scanned at a test site of about 260 m in Beijing, using a Z + F 5016 three-dimensional laser scanner and FSINS3X inertial navigation system. In the experiment, the car with the GNSS and IMU integrated navigation system installed was used to calculate and compare the trajectory and trajectory deviation before and after filter correction. The accuracy of the 3D point cloud generation method based on multi-sensor fusion was experimentally verified. Controlled at 0.037 m, the elevation error was controlled at 0.04 m.

Our proposed method in this study can process the tunnel point cloud data more accurately and quickly and can effectively avoid the deformation of the tunnel and the sharp increase of the error.

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

Funding: This research was funded by the National Natural Science Foundation of China under grant number 42101444 and 42071444, in part by the General scientific research projects of Beijing Municipal Commission of Education under grant number KM202010028012, and Open fund of State Key Laboratory of Rail Transit Engineering Informalization (SKLK22-09).

Data Availability Statement: Not report any data.

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

References

1. Puente, I.; González-Jorge, H.; Martínez-Sánchez, J.; Arias, P. Automatic detection of road tunnel luminaires using a mobile LiDAR system. *Measurement* **2014**, *47*, 569–575. [[CrossRef](#)]
2. Zhou, Y.; Wang, S.; Mei, X.; Yin, W.; Lin, C.; Hu, Q.; Mao, Q. Railway Tunnel Clearance Inspection Method Based on 3D Point Cloud from Mobile Laser Scanning. *Sensors* **2017**, *17*, 2055. [[CrossRef](#)] [[PubMed](#)]
3. Tran, T.H.; Taweep, C. Automated extraction of expressway road surface from mobile laser scanning data. *J. Cent. South Univ.* **2020**, *27*, 1917–1938. [[CrossRef](#)]

4. Liu, S.; Sun, H.; Zhang, Z.; Li, Y.; Zhong, R.; Li, J.; Chen, S. A multi-scale deep feature for the instance segmentation of water leakages in tunnel using MLS point clouds. *IEEE Trans.* **2022**, *60*, 5702716.
5. Sun, H.; Xu, Z.; Yao, L.; Zhong, R.; Du, L.; Wu, H. Tunnel Monitoring and Measuring System Using Mobile Laser Scanning: Design and Deployment. *Remote Sens.* **2020**, *12*, 730. [[CrossRef](#)]
6. Chen, Z.; Li, Q.; Li, J.; Zhang, D.; Yu, J.; Yin, Y.; Lv, S.; Liang, A. IMU-Aided Registration of MLS Point Clouds Using Inertial Trajectory Error Model and Least Squares Optimization. *Remote Sens.* **2022**, *14*, 1365. [[CrossRef](#)]
7. Jung, J.H.; Cha, J.; Chung, J.Y.; Kim, T.I.; Seo, M.H.; Park, S.Y.; Yeo, J.Y.; Park, C.G. Monocular Visual-Inertial-Wheel Odometry Using Low-Grade IMU in Urban Areas. *IEEE Trans.* **2022**, *23*, 925–938. [[CrossRef](#)]
8. Du, L.; Zhong, R.; Sun, H.; Zhu, Q.; Zhang, Z. Study of the Integration of the CNU-TS-1 Mobile Tunnel Monitoring System. *Sensors* **2018**, *18*, 420. [[CrossRef](#)] [[PubMed](#)]
9. Engstrand, A. *Railway Surveying—A Case Study of the GRP 5000*; Division of Geodesy and Geoinformatics Royal Institute of Technology (KTH): Stockholm, Sweden, 2011.
10. Du, L.; Zhong, R.; Sun, H.; Liu, Y.; Wu, Q. Crosssection positioning based on a dynamic MLS tunnel monitoring system. *Photogramm. Rec.* **2019**, *34*, 244–265. [[CrossRef](#)]
11. Han, Y.; Sun, H.; Zhong, R. Three-Dimensional Linear Restoration of a Tunnel Based on Measured Track and Uncontrolled Mobile Laser Scanning. *Sensors* **2021**, *21*, 3815. [[CrossRef](#)] [[PubMed](#)]
12. Sun, J.; Sun, H.; Zhong, R.; Han, Y. Deformation Detection Method of Mine Tunnel Based on Mobile Detection System. *Sensors* **2020**, *20*, 5400. [[CrossRef](#)] [[PubMed](#)]
13. Li, L.; Zhang, X.; Lian, J.; Zhou, Y.; Zhen, W. Semantic SLAM algorithm based on road structural features. *J. Harbin Inst. Technol.* **2021**, *53*, 9.
14. Karam, S.; Vosselman, G.; Peter, M.; Hosseinyalamdary, S.; Lehtola, V. Design, Calibration, and Evaluation of a Backpack Indoor Mobile Mapping System. *Remote Sens.* **2019**, *11*, 905. [[CrossRef](#)]
15. Crassidis, J.L.; Junkins, J.L. *Optimal Estimation of Dynamic Systems*, 2nd ed.; Chapman & Hall/CRC Applied Mathematics & Nonlinear Science: Boca Raton, FL, USA, 2011.
16. Fraser, D.; Potter, J. The optimum linear smoother as a combination of two optimum linear filters. *IEEE Trans.* **1969**, *14*, 387–390. [[CrossRef](#)]
17. Fraser, D.; Potter, J. Massachusetts Institute of Technology, Cambridge, MA, USA. *IEEE Trans.* **1969**, *14*, 387–390.
18. Ryall, T.G. *Maximum Likelihood Estimates of Linear Dynamic System Parameters*; Aeronautical Research Laboratories: Columbus, OH, USA, 1983.
19. Teng, H.; Wang, Y.; Wang, L. Application Analysis of RTS Smoothing Filtering on the Data Post-Processing of BDS/INS Integrated Navigation. *Int. Conf. Mechatron. Autom. Eng.* **2017**, 440–452. [[CrossRef](#)]
20. Xuan, L.; Jing, Y. Application of RTS optimal smoothing algorithm in satellite attitude determination. In Proceedings of the 2011 2nd International Conference on Intelligent Control and Information Processing, Harbin, China, 25–28 July 2011; pp. 978–982.
21. Jiang, Q. *Research on High-Precision Inertial Measurement and Parameter Estimation Method Based on Smooth Filtering*; National University of Defense Technology: Changsha, China, 2017.
22. Titterton, D.; Weston, J. Strapdown Inertial Navigation Technology. *Aerosp. Electron. Syst. Mag. IEEE* **2004**, *20*, 33–34. [[CrossRef](#)]
23. Seube, N.; Picard, A.; Rondeau, M. A simple method to recover the latency time of tactical grade IMU systems. *ISPRS J. Photogramm. Remote Sens.* **2012**, *74*, 85–89. [[CrossRef](#)]
24. Chen, Q.; Zhang, Q.; Niu, X. Estimate the Pitch and Heading Mounting Angles of the IMU for Land Vehicular GNSS/INS Integrated System. *IEEE Trans.* **2020**, *99*, 1–13. [[CrossRef](#)]
25. Li, Q.; Bai, Z.; Chen, B.; Guo, J.; Xin, H.; Cheng, Y.; Li, Q.; Wu, F. GNSS/INS multi-sensor integrated high-speed railway track measurement system. *J. Surv. Mapp.* **2020**, *49*, 569–579.