IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received February 11, 2020, accepted February 27, 2020, date of publication March 3, 2020, date of current version March 12, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2977923

Unmanned Aerial Vehicle Positioning Algorithm Based on the Secant Slope Characteristics of Transmission Lines

CHANG'AN LIU[1,2], YUE SHAO[1], ZIQIANG CAI[1], AND YUEJIE LI[1,3]

[1]Department of Control and Computer Engineering, North China Electric Power University, Beijing 102206, China
[2]Department of Information, North China University of Technology, Beijing 100144, China
[3]Ordos Institute of Technology, Ordos 017000, China

Corresponding author: Yue Shao (shaoyue126069@163.com)

This work was supported in part by the National Natural Science Fund Project under Grant 61105083, and in part by the Central University Basic Scientific Research Business Funds under Grant 2018 ZD06.

ABSTRACT Aiming at the problem of positioning consistency and real-time performance during power inspection of unmanned aerial vehicles (UAV), this paper proposes an UAV positioning algorithm based on the secant slope characteristics of transmission lines. The traditional correlation scan matching (CSM) positioning method is prone to mismatch under the condition of a single geometric feature. Although the inertial measurement unit (IMU) -aided CSM method improves the positioning accuracy, using the IMU for a long time to estimate the position is prone to cumulative errors and increases the computational cost to a certain extent. Our proposed method uses the least square polynomial curve fitting method to parametric reconstruction of the transmission line, which enhances the geometric characteristics of the transmission line and made up the missing part. The secant line of transmission line is extracted and its feature is parameterized to match with the reconstructed transmission line map, and the translation vector and rotation vector of UAV are estimated. Through multiple sets of experiments, it is proved that our algorithm is less than the CSM method and the IMU-aided CSM method in terms of average translation error, average rotation error, and positioning time. Our algorithm not only improves the positioning accuracy but also guarantees the real-time positioning requirements.

INDEX TERMS Cumulative error, point cloud matching, secant slope, transmission line.

I. INTRODUCTION

In the power transmission line patrol operation and maintenance work, UAV based patrol has significant application advantages, and the precise positioning technology of UAV is the basis of the realization of obstacle avoidance, navigation, path planning and other inspection functions. First of all, the precise positioning technology of UAV can make it more accurate to reach each task point, reduce the number of shots of the detected target due to the wrong pose information, and improve the accuracy of information collection; through the precise positioning technology, the drone can realize full autonomous mission flight, which can effectively avoid repeated taking photos or missing shots. At the same time, accurate positioning technology can improve the efficiency of data collection to meet the needs of the mission during the drone cruise.

The associate editor coordinating the review of this manuscript and approving it for publication was M. Jahangir Hossain.

At present, robot positioning technologies are mainly divided into three categories: dead reckoning (DR) methods, signal-based positioning methods, and laser scanning matching methods.

The DR [1] method mainly relies on sensors such as IMU to infer the current pose based on the pose at the previous moment. Using the gyroscope and accelerometer to measure the robot's angular and linear motion parameters, based on Newton's laws of mechanics, the speed, position and attitude of the robot can be calculated. The output frequency of this method is high, which satisfies the real-time performance of positioning, and the short-term positioning accuracy is high. However, the method has accumulated errors, which gradually increase with time, which eventually leads to positioning failure. This method is usually used in combination with a signal-based positioning method or a laser scanning matching method to improve positioning accuracy.

The signal-based positioning method is represented by the Global Navigation Satellite System (GNSS). GNSS [2] is

one of the key technologies for realizing geolocation. This method has improved the accuracy after the real-time kinematic (RTK). RTK [3] is to install the GNSS monitoring receiver on an accurate known base station and calculate the distance correction between the base station and the GNSS satellite. This positioning method can reach the centimeter level. However, RTK has the following problems: base station installation cost is high and power consumption is high; its positioning accuracy is very dependent on the number of satellites; it is susceptible to electromagnetic interference; when it is blocked, it is prone to signal loss and cause positioning failure.

The laser scanning matching method uses the observed features to match the database features and stored features to obtain the robot pose. This method requires a positioning map to be made in advance. In recent years, positioning methods based on laser scanning matching have occupied the main position in the research of UAV positioning.

Motivated by the rapid development of robot positioning technology and the large demand for inspections in the power industry, many researchers have conducted research on the location of UAV during power inspections. When the robot starts to move from an unknown position, it will position itself according to the position estimation and the existing environment map during the movement. At the same time, an incremental map will be built on the basis of its own positioning to realize the autonomous positioning and navigation of the robot. However, the cumulative error caused by long-term pose estimation is inevitable, and there will be a large deviation between the constructed positioning map and the real map. Therefore, we have improved the existing method and proposed an UAV positioning method based on the secant slope characteristics of transmission lines. We mainly study the consistency of UAV in power inspection. Based on the known environment map, the offline optimization of the environment map and the positioning of UAV under the known environment map are explored. The primary contribution of the paper can be summarized as follows:

1) PARAMETRIC RECONSTRUCTION OF THE TRANSMISSION LINE

The problem of data loss and single characteristic of the transmission line are solved by parameter reconstruction. Firstly, the geographic information model of power inspection is established, the transmission line environment with similarity is modeled. Then, analyzed the geometric characteristics of the transmission lines, and parameterized the transmission lines by least square polynomial curve fitting method. Finally, the performance of the proposed parametric reconstruction method is verified by experiment. A method for separately storing point cloud data of power poles and transmission lines is proposed, which reduces memory consumption by more than a thousand times compared with the original storage method.

2) FEATURE MATCHING ALGORITHM BASED ON SECANT SLOPE

Aiming at the high similarity of transmission line environment, if we only use IMU integral to estimate the position of UAV, there will be a large cumulative error of the UAV pose. Therefore, we propose a localization algorithm based on the secant slope characteristics of the transmission lines. First, according to the laser scanning data, extract the slope characteristics of the transmission line after IMU attitude correction. Then, extract the slope of the secant line, match with the reconstructed transmission line map, which can improve the accuracy of position estimation by optimizing the translation vector and yaw angle. Finally, several experiments are carried out, the effect of our method is compared with CSM method and IMU-aided CMS method. The results show that the proposed method has higher accuracy and real-time performance.

II. RELATED WORK

The localization methods for laser scanning matching are basically divided into three categories [4]: filtering method; iterative and gradient descent method; and probabilistic grid method.

The method based on the filter mainly uses the recursive Bayesian principle to estimate the current pose of the robot according to the control input information, prior pose information and observation information. Common filter algorithms are: Extended Kalman Filter (EKF) [5], Unscented Kalman Filter (UKF) [6], Particle Filter (PF) [7]–[10]. The main emphasis of the filter is time and increment characteristics, this method is also called online filtering method. The EKF method can obtain better state estimation under the condition that the system is less nonlinear and satisfies the Gaussian distribution [11]. In this method, robot pose tracking is regarded as a linear problem with simple calculation and fast speed. However, in practical application, due to the highly nonlinear environment of the system, the error caused by the linearization of the system accumulates over time, resulting in positioning failure [12]. Hong *et al.* [13] used the three-axis attitude determination (TRIAD) algorithm to obtain the directional cosine matrix (DCM) and proposed an EKF algorithm using the three directional cosines of DCM as the measurement vector to reduce the cumulative error. Jwo and Tseng [14] proposed an EKF method based on interactive multiple model (IMM), which uses IMM estimation as the input of multiple parallel filters to match different motion modes, and finally performs weighted average to assist EKF estimation, improves the positioning accuracy of the robot. To solve the problem of robot positioning stability, Zhou *et al.* [15] proposed an UKF based pose estimation and tracking method to stabilize the nonlinear system. Using Lyapunov theory, the asymptotical stability of the system was guaranteed, and the effectiveness and robustness of the algorithm were greatly improved. However, due to its sensitivity to data noise, the pose tracking performance is still insufficient. In addition, Effati and Skonieczny [16] carried

out a large number of UKF positioning and EKF positioning experimental results and found that in most cases, the UKF convergence rate is fast, and in some cases only the UKF convergence. However, how well the traditional PF algorithm solves the posterior probability of the system depends on the number of samples. It has poor real-time performance in the localization process of complex environments and it is difficult to meet the task requirements.

The current mainstream iterative and gradient descent methods are: Iterative Closest Points (ICP) and Normal Distribution Transformation (NDT). The implementation principle of the ICP algorithm is based on the least square optimal registration method. By repeatedly selecting corresponding point pairs, the optimal rigid body transformation is calculated until the convergence accuracy requirements for correct registration are met. The traditional ICP algorithm consumes more computing time as the environment grows during the matching process, making it difficult to guarantee real-time performance during the positioning process. To solve this problem, R. Tiar et al. proposed a local iterative closest points-simultaneous localization and mapping (ICP-SLAM) algorithm based on local environment division. By applying the ICP algorithm to the local map, this method allows the calculation time to be within a fixed time range, and overcomes the influence of the calculation time on the positioning results [17]; Li S et al. proposed an improved ICP algorithm based on k-dimensional tree (k-d tree), which uses k-d tree to store, manage and search point cloud data, which greatly improves the calculation efficiency of traditional ICP algorithm [18]; Most scanning matching algorithms need to find corresponding points or lines between the characteristic points or lines during the matching process, which will consume a certain amount of calculation time. Biber and Strasser [19] proposed the normal distributions transform (NDT) algorithms to replace them, subdivide the two-dimensional plane into cells, and construct the normal distribution of multi-dimensional variables based on laser scanning data. Because the performance of NDT algorithm is related to cell size, it is difficult to achieve the optimal selection of cell size. Ulaş and Temeltaş [20] proposed a multi-layered normal distribution transform method (MLNDT), this method establishes a multi-level planar structure and divides it by cells of different sizes, which effectively solves the effect of cell size selection on the NDT algorithm.

In the positioning process, due to the measurement noise of the sensor and the uncertainty of the environment, the probabilistic grid method is usually used to estimate the pose of the robot. The probabilistic localization method is implemented based on the Bayesian filter [21], which uses Bayes' theorem to provide a recursive solution formula for calculating the probability density function. Markov localization uses the approximate grid method to discretize all possible positions and postures of the robot, in which the weight of each cell represents the probability of the robot at that position, and the localization process is mainly to update the weight of each cell. Markov positioning is a state-space sampling method that can represent any probability distribution. It is not sensitive to measurement noise and has high reliability; however, since the weights of these cells are updated after each sensor measurement, the time and space complexity of the process is relatively high, and a known state space is required, so this method is usually only used in indoor environments [22]. In order to solve this problem, Olson [23] proposed a correlation scan matching (CSM) algorithm. Through the grid division of space, the Gauss distribution is used to evaluate and diffuse the grid, and the transformation with the highest score is used as the pose transformation of the robot. This method achieves higher positioning accuracy and good robustness at the cost of additional time. In order to make the calculation more efficient, Olson proposed a method using multi-resolution [24]. This method first uses a low-resolution grid to find a candidate transformation matrix, and then uses the result as an initial value for high-resolution grid calculation. Because when searching with a low-resolution grid, the speed is faster than high-resolution. After obtaining the transformation matrix through low-resolution, perform preliminary transformation on the data, and then use high-resolution optimization to save a lot of computing time. Experiments show that the calculation speed of this method is 10 times that of the single high-resolution method.

III. MAP OPTIMIZATION

A. GEOGRAPHIC INFORMATION MODEL OF ELECTRICITY INSPECTION

Whether using the laser data matching method for positioning in a 2D environment or a 3D environment, it is necessary to pre-made an environment map. The data acquired by the laser LIDAR are matched with the pre-made map to obtain the position and posture, and then the actual position and posture of the robot are calculated by the external parameters between the laser LIDAR and the robot.

The point cloud data used in this study was collected using velodyne 16-line LIDAR in a high-voltage electrical tower environment. The acquisition of environmental information by UAV through a laser scanning device typically results in point cloud data of uneven density. Due to factors such as changes in the surface characteristics of the measured object and the mismatching of the registration process of the point cloud data, it is hard to avoid some noise points. In addition, when the power towers and the transmission lines are measured by laser scanning, there are always a large number of outliers far away from the measured object, and the measurement error in the process of acquiring data will produce sparse outliers. These factors may lead to the failure of point cloud data in the matching process, so we build environment map by combining scan matching and closed-loop detection [25], and further optimize the map by point cloud optimization software. Figure 1 is the point cloud map of transmission line and power tower after preliminary optimization of the constructed map.

According to the characteristics of the environmental map constructed in the electric power inspection, it can be divided

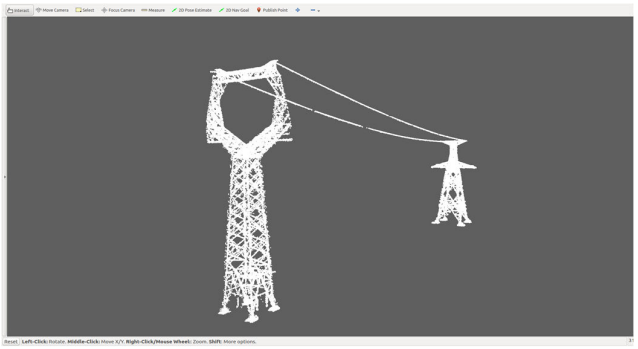


FIGURE 1. Environmental map of electric power inspection.

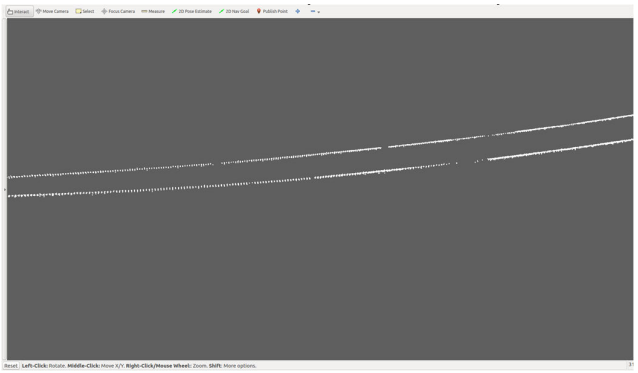


FIGURE 2. Enlarged view of the transmission lines.

into two parts: namely the power towers and the transmission lines. We can find that the features of the power towers are rich, so the results obtained by the laser scanning matching algorithm can meet the positioning requirements.

Figure 2 is an enlarged view of the transmission lines, we can see that geometric features of the transmission lines are single, some of the maps are missing, which is easy to cause mismatch, so it is difficult to meet the positioning requirements. We consider reconstructing the missing data to enhance the geometric features, and further correct the position of UAV to improve the consistency of positioning. Therefore, we parameterized the environment of the transmission line and stored it separately from the power towers.

B. PARAMETRIC RECONSTRUCTION OF THE TRANSMISSION LINE

The transmission line is made of metal material. In winter, the metal shrinks when the weather is cold. If the length of the wire is equal to the distance between the two towers, it will be broken by the contraction force. Therefore, the length of the transmission line is usually longer than the distance between the two power towers, showing a naturally drooping posture. It can be considered as a line segment between two points, and the length of the line segment is longer than the distance between the two points. The physical model droops naturally due to gravity, similar to a parabolic model, so we reconstructed the transmission lines according to this characteristic. In this study, the least square polynomial curve fitting method [26] is utilized to represent the whole

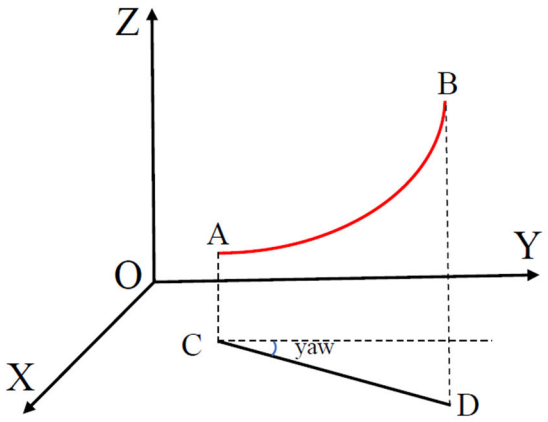


FIGURE 3. An intercepted transmission line.

transmission line to compensate for the missing pieces and enhance the geometry features.

Firstly, we intercepted a section of the transmission line and selected the endpoints of the line by distance. Then calculate the angle between the projection of the line in the XOY plane and the Y axis, the angle is yaw, as is shown in Figure 3.

The red curve AB in the Figure 3 is the intercepted transmission line. Let $\overrightarrow{CD}$ be the line direction vector after the curve projection, $\vec{n} = (0, 1, 0)$ be the y-axis direction vector. Then the angle is obtained:

$$yaw = \arccos\left(\frac{\overrightarrow{CD} \cdot \vec{n}}{\left|\overrightarrow{CD}\right| \cdot |\vec{n}|}\right) \tag{1}$$

Line point cloud Z is transformed by the rotation matrix R, and the cross section of the transformed point cloud Z' is parallel to the YOZ plane:

$$R = \begin{pmatrix} \cos(yaw) & 0 & -\sin(yaw) \\ 0 & 1 & 0 \\ \sin(yaw) & 0 & \cos(yaw) \end{pmatrix},$$
$$Z\prime = R \cdot Z \tag{2}$$

Fit the two-dimensional point cloud data after projection, and set the fitting polynomial as:

$$f(x) = \Sigma_{k=0}^{n} \alpha_k \phi_k(x) \tag{3}$$

where: α_k is the polynomial coefficient, ϕ_k is the function corresponding to each coefficient.

Equation (4) is the sum of the squares of the deviations from each point to the curve:

$$S^2 = \Sigma_{i=1}^{m} [\Sigma_{k=0}^{n} \alpha_k \phi_k(x_i) - f(x_i)]^2 \tag{4}$$

The fitting process is transformed into finding the minimum of the equation (4). Therefore, the partial derivatives of the coefficients $\alpha_0, \alpha_1, \ldots, \alpha_k$ are respectively made equal to zero:

$$\frac{\partial S}{\partial \alpha_k} = -2 \sum_{i=1}^{m} \phi_k(x_i) \times [\alpha_0 \phi_0(x_i) + \alpha_1 \phi_1(x_i) + \ldots + \alpha_n \phi_n(x_i) - f(x_i)] = 0 \tag{5}$$

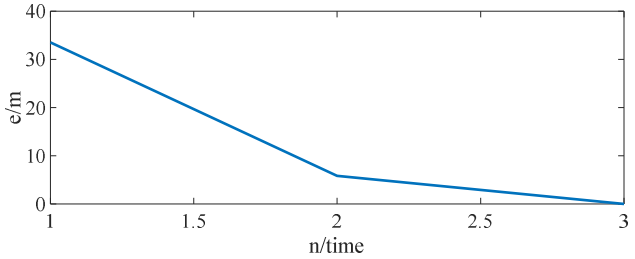


FIGURE 4. Error-iteration curve.

Converting these derivations from equation (5) into a matrix form:

$$\underbrace{\begin{bmatrix} 1 & \phi_1(x_1) & \cdots & \phi_k(x_1) \\ 1 & \phi_1(x_2) & & \phi_k(x_2) \\ \vdots & & \ddots & \vdots \\ 1 & \phi_1(x_n) & \cdots & \phi_k(x_n) \end{bmatrix}}_{X} \underbrace{\begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_n \end{bmatrix}}_{A} = \underbrace{\begin{bmatrix} f(x_1) \\ f(x_2) \\ \vdots \\ f(x_n) \end{bmatrix}}_{Y} \tag{6}$$

That is $X \cdot A = Y$. Equation (7) is the coefficient matrix A:

$$A = (X^T \cdot X)^{-1} \cdot X^T \cdot Y \tag{7}$$

According to the characteristics of the transmission line model, $\varphi_k(x)$ can be taken for x^k, $k \in (0, 1, 2\ldots)$. Define the error threshold as E_T. The fitting steps are as follows:

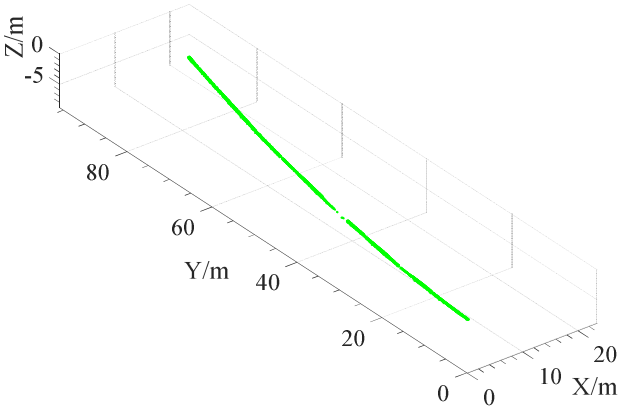
(a) Rotating the original data by equations (1) and (2), parallel to the *YOZ* plane;

(b) Taking $E_T = 0.001, k = 0, 1, 2, \ldots$,curve fitting is performed by equations (3), (4), (5), (6), and (7) until the curve deviation S converges to less than E_T, resulting in a final curve.

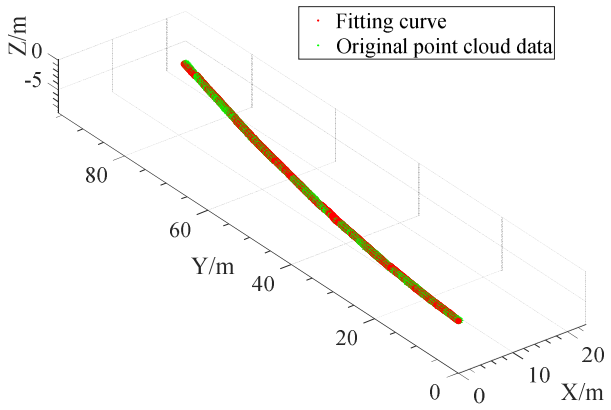
Figure 4 shows the number of the iterations and the error curve during the fitting process. In the figure, n is the number of iterations, e is the curve error (unit: meter) of each iteration, that is, when $n = 3, k = 2$, the curve deviation S converges to less than E_T.

Figure 5 shows the comparison of the fitted curve with the original point cloud data. In Figure 5 (a), the green part is the original point cloud data, the data is discontinuous in some places and partially missing. In Figure 5 (b), comparing the results before and after the fitting: we can clearly see that the green point cloud data is mixed with many red parts. These red parts are the missing point cloud data made up by the least square polynomial curve fitting method, the fitting curve is smooth and continuous. In Figure 5 (c), comparison of projection in YOZ plane before and after curve fitting. After fitting the point cloud data, the missing segment is effectively compensated.

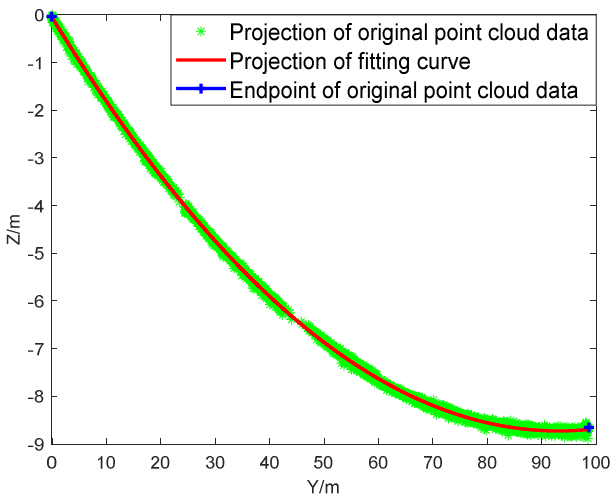
In the data we collected, the number of point cloud on the transmission line was 14367. The coordinates of each point in three-dimensional space are represented by three float type parameters: X, Y, Z, so each point cloud occupies 12 bytes of memory; The memory consumption of the point cloud data storage for the entire transmission line is: $12 \times 14376 = 164.42KB$.



(a) Original Point Cloud Data



(b) Least Square Polynomial Fitting Curve



(c) Comparison of Projection Effect before and after Fitting

FIGURE 5. Comparison of original point cloud data and fitting curve.

After parameterized reconstruction of the transmission line by curve fitting, we proposed a new data structure to store a single transmission line:

(1) The coefficients of the curve polynomial are generally more than 3 and less than 10 parameters of the double type, that is, occupying 24-80 Byte memory.

(2) Endpoints of the transmission line: each transmission line has two endpoints, the coordinates of each of them are in the form of (X, Y, Z), that is, occupying 24 Byte memory.

(3) Endpoints of the fitting curve: each fitting curve has two endpoints, and the coordinates of each endpoint are in the form of (Y, Z), that is, occupying 16 Byte memory.

It can be obtained by calculation that the memory consumption of our method is: $64Byte - 120Byte$. The memory consumption is reduced by more than 1000 times compared with the original storage mode. At the same time, the memory consumption of the original storage mode increases with the increase of the number of point cloud data and the number of transmission lines. However, our storage method can guarantee the memory consumption in a very small range, which is not affected by the number of point cloud data, and only increases with the number of transmission lines.

IV. SECANT SLOPE FEATURE MATCHING ALGORITHM

A. EXTRACTION OF TRANSMISSION LINE AND SECANT ENDPOINTS

Since there are at least two transmission lines across the two towers, in order to extract the endpoints of any of the transmission lines, it is necessary to extract the line from the sensing data first. In this paper, point-to-point distance propagation method [27] is used to extract a single transmission line in the sensing data. Take the first point as the initial point, set the distance threshold between the points, propagate the distance to its neighboring points, and the point that meets the distance threshold is the next initial point, until all of the data is traversed, finally a transmission line is obtained. The points corresponding to the minimum and maximum value in the Y direction is taken as the initial endpoint of the secant line, traverse the neighboring points in the remaining part of the extracted line respectively, and calculate the center of gravity of all neighboring points as the final secant endpoint. The specific algorithm flow is as follows:

(1) The pre-processed sensing data in section III is stored in the array $\boldsymbol{scan\_data}[\boldsymbol{n}]$, where n is the number of point clouds in the data and $scan\_data[1]$ is the initial propagation point $\boldsymbol{temp}$; At the same time, the dynamic array $\boldsymbol{extract\_line}$ and $\boldsymbol{other\_data}$ are created to record the extracted point cloud data and the remaining point cloud data respectively. Set the distance threshold as $\boldsymbol{T\_distance}$.

(2) Calculate $\boldsymbol{distance}$ between point $\boldsymbol{scan\_data}[\boldsymbol{i}](\boldsymbol{i} \neq \boldsymbol{1})$ and propagation point $\boldsymbol{temp}$. If $\boldsymbol{distance} < \boldsymbol{T\_distance}$, $\boldsymbol{scan\_data}[\boldsymbol{i}]$ is stored in dynamic array $\boldsymbol{extract\_line}$, set $\boldsymbol{temp} = \boldsymbol{scan\_data}[\boldsymbol{i}]$. Otherwise, $\boldsymbol{scan\_data}[\boldsymbol{i}]$ is stored in dynamic array $\boldsymbol{other\_data}$.

(3) Repeat step (2) until all points are iterated through. The data stored in $\boldsymbol{extract\_line}$ is the extracted single transmission line data.

(4) Traverse to find the minimum and maximum points in $\boldsymbol{extract\_line}$ as the initial endpoints of the transmission line: $\boldsymbol{point\_A}$ and $\boldsymbol{point\_B}$.

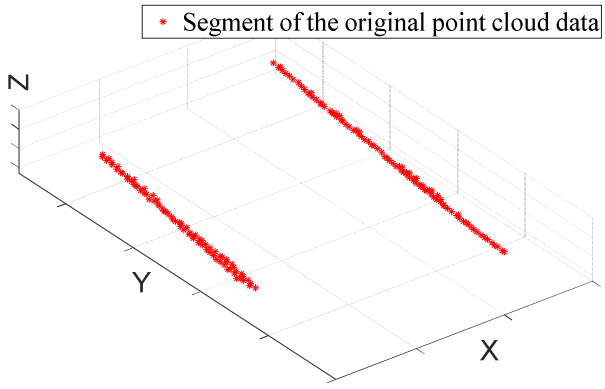


FIGURE 6. Original laser sensing data.

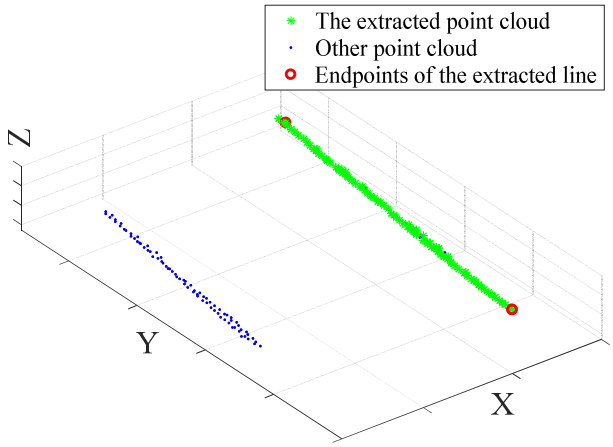


FIGURE 7. Effect after extraction of line and endpoints.

(5) Simultaneously traverse to find the adjacent points of $\boldsymbol{point\_A}$ and $\boldsymbol{point\_B}$ in the array of $\boldsymbol{extract\_line}$, calculate the gravity center of its adjacent points as the final secant endpoints, and record their serial numbers as $\boldsymbol{A\_num}$ and $\boldsymbol{B\_num}$.

Figure 6 shows the original sensing data.

Figure 7 shows the effect of line and endpoint extraction. The green part is the point cloud data of a transmission line extracted by the point-to-point distance propagation method [27]. The red circle is the endpoints of the secant line. The blue part is the point cloud data of another transmission line.

B. SECANT SLOPE CHARACTERISTICS PARAMETERIZATION

In order to make the matching algorithm more efficient, the three-dimensional problem is simplified into the geometric features of the secant in the two-dimensional plane, and then parameterized into one-dimensional slope features for matching, which greatly reduces the time complexity of the algorithm.

The parameterization of secant slope features can be separated into two parts: two-dimensional plane projection and secant slope calculation.

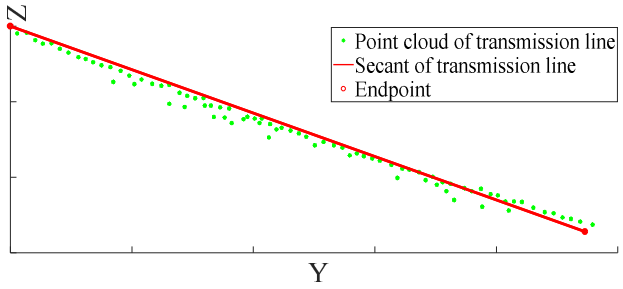


FIGURE 8. Secant line after data projection.

Let $M_i(i = 0, 1, \ldots, n)$ be the two-dimensional data, which is obtained by projecting the extracted point set P_j onto a two-dimensional plane. M_i^y is the horizontal axis coordinate after projection of point i, M_i^z is the vertical axis coordinate after projection of point i. Assuming that P_A is the initial point:

$$\begin{cases} M_i^y = \sqrt{(P_i^x - P_A^x)^2 + (P_i^y - P_A^y)^2} \\ M_i^z = P_i^z \end{cases} \tag{8}$$

Equation (9) is the corresponding secant slope k and the horizontal axis distance d between two secant endpoints:

$$\begin{cases} k = \dfrac{M_A^z - M_B^z}{M_A^y - M_B^y} \\ d = |M_B^y - M_A^y| \end{cases} \tag{9}$$

Figure 8 is the secant feature extracted after data projection. The green part is the point cloud data of the transmission line, the red line is the secant of the extracted transmission line, and the red circle is the endpoints of the secant.

C. ERROR MODLE

As is shown in Figure 9, the red line is the fitting curve. Set the endpoints of target secant as T and T', and the distance of horizontal direction between T and T' is d. The endpoints at both ends of the curve are A and B. According to the horizontal axis distance d, the other endpoints A' and B' of the respective secant can be calculated, as well as the respective center points M1 and M2, and the center points M' between M1 and M2 can be obtained. Take any point E on the curve, and point E' whose horizontal axis distance from point E is d is the predicted endpoint of the secant.

Based on the fact that the secant slope of any two points on the curve is unique when the distance between horizontal axis is fixed, as shown in equation (10), the error can be changed to the angle between the target secant and the predicted secant. Equation (11) is the predicted secant error model:

$$\begin{cases} k_{EE'} = (E'_z - E_z) \Big/ (E'_y - E_y) \\ k_{TT'} = (T'_z - T_z) \Big/ (T'_y - T_y) \end{cases} \tag{10}$$

$$e = \arctan[(k_{EE'} - k_{TT'}) \Big/ (1 + k_{EE'} \times k_{TT'})] \tag{11}$$

In the equation above, $k_{EE'}$ is the predicted secant slope, $k_{TT'}$ is the target secant slope, and e is the predicted error.

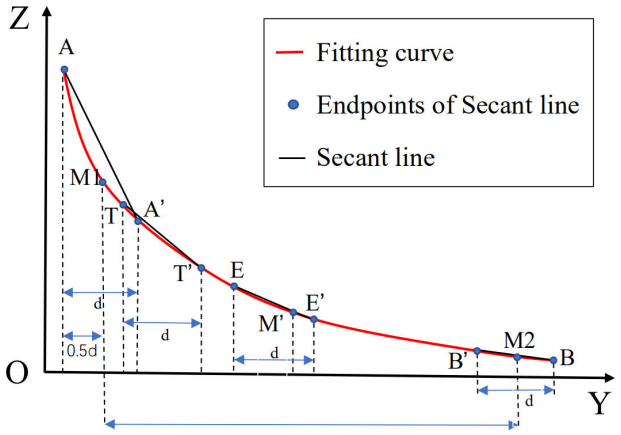


FIGURE 9. Predicted secant and target secant.

D. SECANT SLOPE FEATURE MATCHING ALGORITHM

Select one endpoint of the curve as the starting prediction secant to search the target secant slope. The equation for updating the predicted secant slope can be obtained as follows:

$$k_{EE'} = \frac{[(E'_z - \Delta z) - (E_z - \Delta z)]}{\left[\left(E'_y - \Delta y\right) - \left(E_y - \Delta y\right)\right]} \tag{12}$$

where: Δy, Δz are the update steps in direction Y and direction Z, respectively. Until the error e obtained in equation (11) converges to a certain range and the search is completed.

This method has a great dependence on the selection of updating step, which directly affects the efficiency, accuracy and convergence of the algorithm. The secant slope in the transmission line curve also has the characteristic of increasing or decreasing monotonously when the distance between the horizontal axis is fixed, so the dichotomy method [28], [29] is used to quickly search the secant slope. Equation (13) and (14) are the updated equations for predicting the secant slope:

$$k_{EE'} = \frac{[(M'_z + 0.5d) - (M'_z - 0.5d)]}{\left[\left(M'_y + 0.5d\right) - \left(M'_y - 0.5d\right)\right]} \tag{13}$$

$$\begin{cases} M'_y = (M1_y + M2_y)/2 \\ M'_z = (M1_z + M2_z)/2 \end{cases} \tag{14}$$

where: d is the distance difference of the target secant along the horizontal axis. Calculate the error e by the equation (11), (13), and (14), until it converges to a certain range, ending the search. This method does not have to choose the updating step, and the accuracy and computational efficiency are higher than the former.

The flow of secant slope feature matching algorithm based on dichotomy is as follows:

1) Initial endpoints A, A', B, B', error threshold E_T;
2) Take points A and B at each end of the curve. Based on the transverse axis distance d, calculate the corresponding points A' and B',and then obtain their respective

central points $M1$ and $M2$,using the above points as the initial boundary points;

3) Let $k_{TT'}$ be the target secant slope. The slope $k_{AA'}$ and $k_{BB'}$ can be calculated respectively. If $k_{TT'}$ is not within the $k_{AA'}$ and $k_{BB'}$ intervals, the match fails; If $k_{AA'}$ and $k_{BB'}$ are equal to $k_{TT'}$, then the matching ends, A and A' or B and B' are the target secant endpoints, otherwise the next step is executed;
4) The coordinate point M' corresponding to the curve is obtained by calculating the center point in the horizontal directions of points $M1$ and $M2$, which is used as the center point of the predicted secant endpoint. At the same time, the new endpoints E and E' of the predicted secant are calculated, and the secant slope $k_{EE'}$ is calculated. If $k_{EE'}$ equals $k_{TT'}$, the matching ends, and the predicted secant endpoint is the target secant endpoint; If $k_{EE'}$ is smaller than $k_{TT'}$, set M'e' to $M1$, and re-execute step 3; if $k_{EE'}$ is greater than $k_{TT'}$, set M'e' to $M2$ and re-execute step 3.
5) Step 3 is iterated until the error is less than the given error threshold E_T. Finally, the predicted secant endpoints E and E' are obtained.

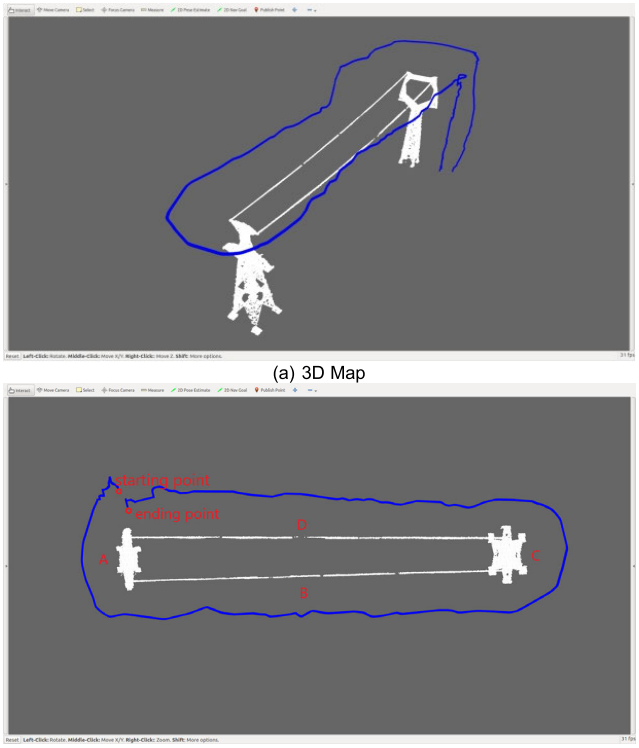
V. EXPERIMENTS

In this section, we have conducted several sets of experiments to evaluate the performance of our proposed algorithm, compared with CSM method [23] and IMU-aided CSM method [30]. All of the experiments are implemented in ubuntu16.04, ROS kinetic and MATLAB R2017a, and conducted on a PC with an Intel i5 processor having 3.4GHz and 32.0 GB of memory. Parameters setting: E_T is 0.1. Due to the limitation of conditions, no field test was conducted, but we used the data playback function of ROS. The speed of data playback is the same as that of data collection, so the whole experiment can simulate the process of real-time collection and processing. In the process of mapping, we first need to run the collected point cloud data. The acquisition frame rate of LIDAR is more than 10 frames per second, so when the processing time of each frame is less than the sampling interval of lidar, that is 0.1s, the real-time processing level can be reached.

In Sec V-A, we use the data collected by the velodyne 16-line LIDAR to construct an environment map as the real path. In Sec. V-B, we compare the results of three algorithms using 30 frames of data matching. In Sec. V-C, we use three algorithms to build the map and compare the respective translation errors, rotation errors, and calculation time. In Sec. V-D, the effects of different flight speeds on the performance of each algorithm are compared.

A. 3D MODEL CONSTRUCTION

Based on the point cloud data obtained by LIDAR, the environment map is constructed by the combination of scanning matching and closed-loop detection [25]. The blue track is the real path with an accuracy of 0.05m, as shown in Figure 10 (a), (b).



(a) 3D Map

(b) Top View

FIGURE 10. Environmental map of electric power inspection.

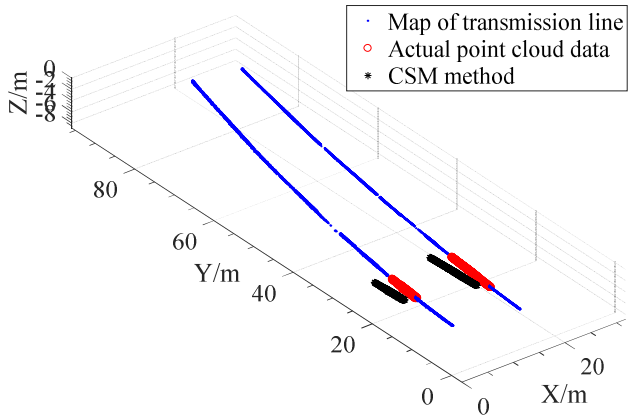
The starting point, ending point, and A, B, C, and D, four areas of the UAV's movement are marked in Figure 10 (b). Among them, A and C are high-voltage tower areas with abundant features, B and D are transmission line areas with single features.

In order to verify the performance of the algorithm proposed in this paper, the off-line positioning experiments are carried out with CSM method and IMU-aided CSM method respectively. None of the above three methods used closed-loop detection to optimize the location results.

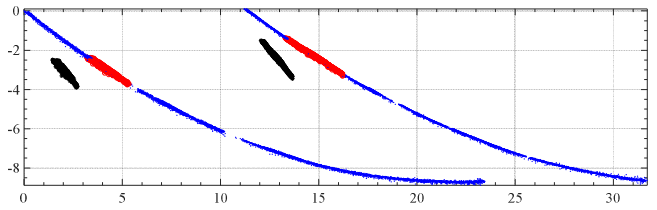
B. CONSTRUCTION OF SUB-MAPS

In order to see the comparison results more clearly, we selected 30 frames of data during the positioning process, used three algorithms (CSM method, IMU-aided CSM method, our method) to construct transmission line submaps, and compared them with the real path constructed in Sec. V-A. Figures 11 (a), (b), (c) are three-dimensional views of the matching results using the CSM method. Figures 11 (d), (e), (f) are three-dimensional views of the matching results using the IMU-aided CSM method. Figures 11 (g), (h), (i) are three-dimensional views of the matching results using the method we proposed. The red part is 30 frames of real point cloud data (30-frame sub-map corresponding to the blue transmission line), the black part is the sub-map obtained by matching with three algorithms.

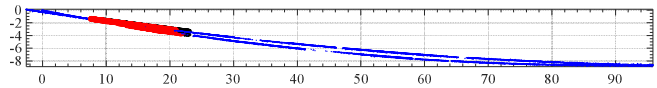
It can be seen from Figure 11 (a), (b) and (c) that CSM method (in black) has large angle deviation and translation deviation compared with the real path (in red).



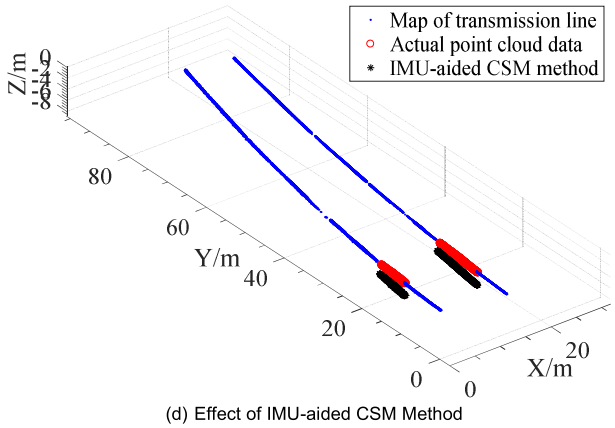
(a) Effect of CSM Method



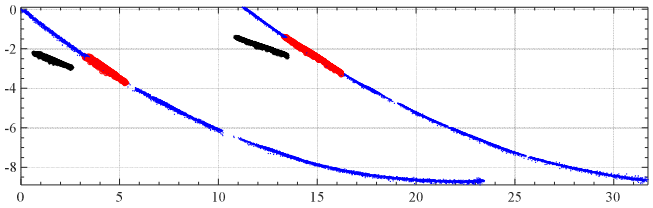
(b) View of CSM Method in ZOX Plane



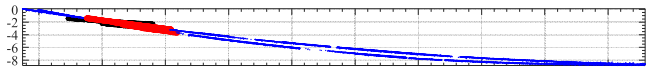
(c) View of CSM Method in ZOY Plane



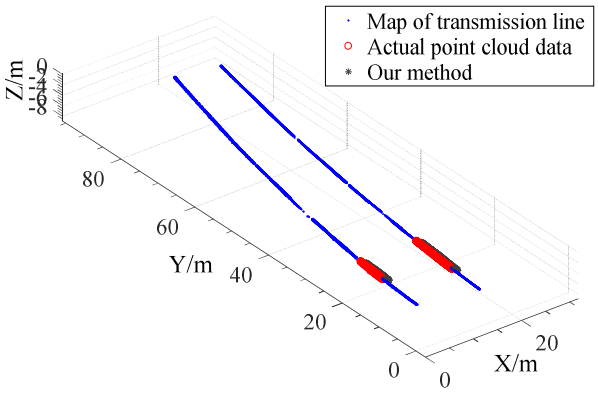
(d) Effect of IMU-aided CSM Method



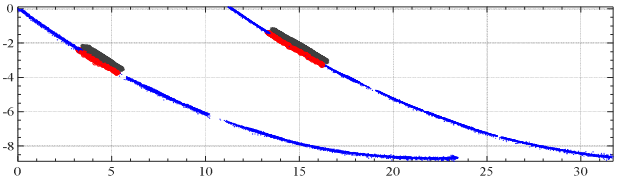
(e) View of IMU-aided CSM Method in ZOX Plane



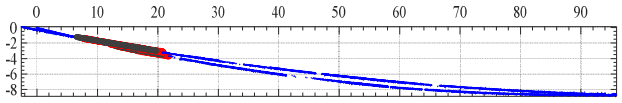
(f) View of IMU-aided CSM Method in ZOY Plane



(g) Effect of Our method



(h) View of Our Method in ZOX Plane



(i) View of Our Method in ZOY Plane

FIGURE 11. **Comparison of our method with CSM method and IMU-aided CSM method.**

In Figure11 (d), (e) and (f), although the rotation deviation of IMU-aided CSM method is smaller than that of CSM method, it results in a larger translation error. Using IMU to estimate the position and attitude can improve the positioning accuracy to a certain extent, but there is accumulated error in using IMU to calculate the track for a long time, which is easy to lead to positioning failure with the accumulation of time. Using our improved method for matching, the generated sub-graph (Figure 11 (g), (h) and (i)) can basically coincide with the real point cloud data, which is closer to the true position of the transmission line and improves the consistency of positioning.

In order to quantify the accuracy and computational cost of the algorithm, we use the method in [31] to calculate the average translation error (ATE), average rotation error (ARE), and time of calculation and sub-map building (t) of the three algorithms for pose estimation, as shown in Table 1.

From Table 1, it can be seen that the proposed method is superior to the CSM method and IMU-aided CSM method in terms of accuracy and time consumption. Among

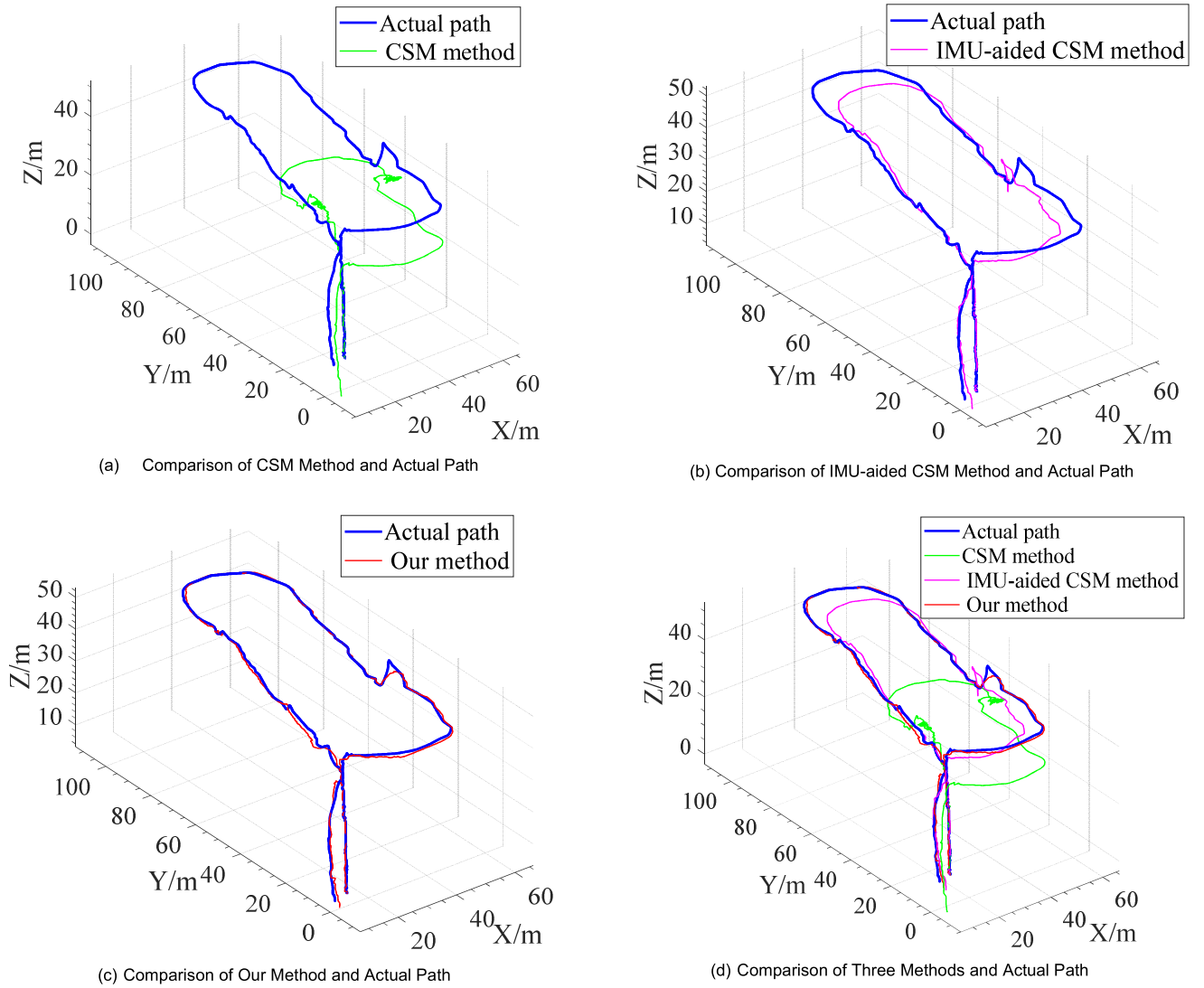


FIGURE 12. 3D view of track results.

TABLE 1. Algorithm performance evaluation.

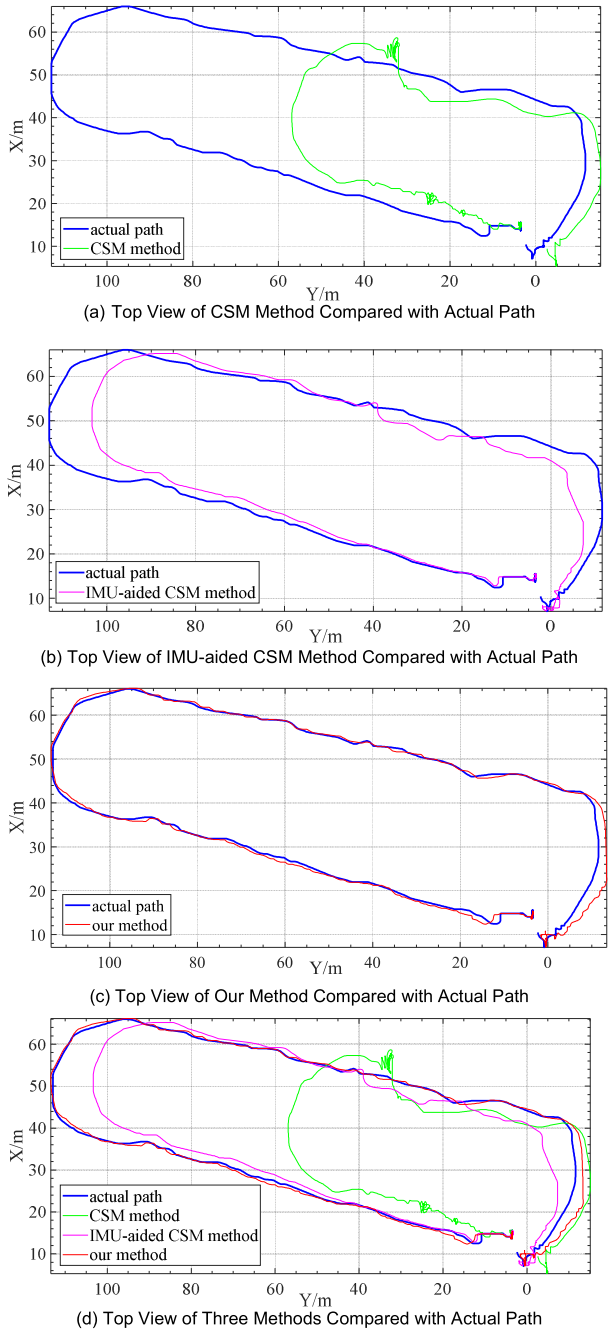
Algorithms	ATE/m	ARE/°	t/s
CSM method	1.795	3.486	1.276
IMU-aided CSM method	2.829	1.223	1.487
The method we proposed	0.483	0.127	0.032

them, ATE is 26.9% of CSM method,17.07% of IMU-aided CSM method; ARE is 3.64% of CSM method, 10.38% of IMU-aided CSM method. It can be seen from the above table that our method only takes 0.032 seconds to process 30 frames of data and build graphs, which means that during the use of the method in this paper, using the secant slope feature can quickly search for the correct position and pose, and meet the real-time requirements of UAV positioning.

C. CONSTRUCTION OF POSITIONING MAP

We have carried out further experiments to compare the effect of three methods of positioning and real-time map building, respectively using three algorithms to let the UAV follow the trajectory of Sec. V-A at a speed of 10km / h. Figure 12 (a), (b), (c), (d) are the three-dimensional maps constructed using three algorithms, respectively. Figure 13 (a), (b), (c), (d) are their top views.

It can be seen from Figure 12 and Figure 13 that the map build by CSM method completely deviates from the real path when it passes through area B (transmission line environment), that is, the positioning fails in the process of area B, but the positioning results are similar with the real path when it passes through area A and area C, so it can be considered that the CSM method is prone to mismatch in the case of single feature and high environmental similarity; IMU-aided CSM positioning method, through the introduction of IMU integral



(a) Top View of CSM Method Compared with Actual Path

(b) Top View of IMU-aided CSM Method Compared with Actual Path

(c) Top View of Our Method Compared with Actual Path

(d) Top View of Three Methods Compared with Actual Path

FIGURE 13. Top view of track results.

to modify the positioning results of CSM method, compared with CSM method, has some improvement, but there is still some deviation in area B and D; Our positioning method is based on IMU-aided CSM positioning, using Euclidean distance between frames to extract key frames every certain interval, using secant slope matching method in the extracted key frame interval to quickly adjust the UAV's position and attitude, and the final positioning results are closer to the real trajectory. Experimental results show that the proposed method is relatively stable and consistent.

we calculate the average translation error (Table 2), average rotation error (Table 3) in areas A, B, C, and D,

TABLE 2. Comparison of average translation errors of three algorithms in different areas.

Algorithms	ATE/(m)			
	A	B	C	D
CSM method	2.228	40.255	3.265	52.448
IMU-aided CSM method	1.852	2.333	3.229	3.063
The method we proposed	1.724	0.255	0.552	0.198

TABLE 3. Comparison of average rotation errors of three algorithms in different areas.

Algorithms	ARE/ (°)			
	A	B	C	D
CSM method	2.802	19.588	2.369	11.957
IMU-aided CSM method	1.755	3.223	1.985	3.933
The method we proposed	1.003	0.996	0.335	0.856

TABLE 4. Comparison of calculation and mapping time of three algorithms.

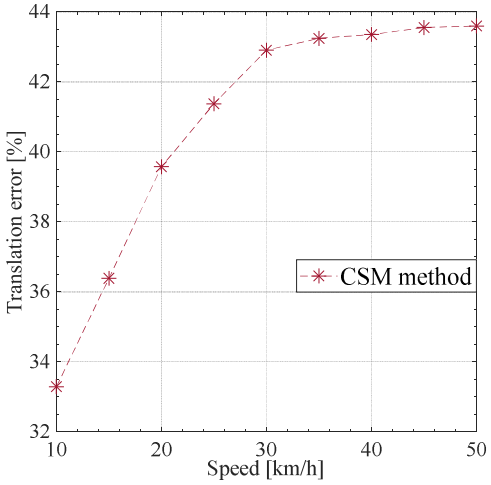
	CSM method	IMU-aided CSM method	The method we proposed
t/s	821.738	1036.461	711.625

Table 4 shows the time of calculation and map building of three algorithms for pose estimation.

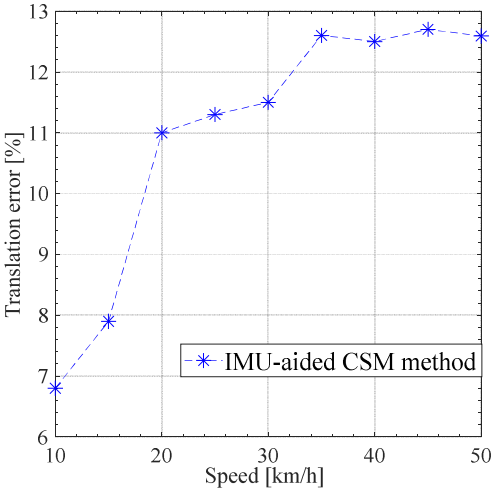
In Table 2, we can see that in area A, the translation errors of the three algorithms are not very large, about two meters. However, in area B, there is a big deviation between the map created by CSM and the real path. This is because the environmental difference of transmission line is too small, and there is no easy to distinguish the environmental characteristics, so CSM method is easy to fall into local extremum. The maximum translation error of the map constructed by CSM method is about 40 to 50 meters. Since IMU is used to estimate the position and attitude of the IMU-aided CSM method, the positioning accuracy of the IMU-aided CSM method is much better than that of the CSM method, but the long-term use of IMU still results in a translation error of 2 to 3 meters. Using our method, no matter in the transmission line area or power tower area, the deviation between the built map and the real path is very small.

As shown in Table 3, the CSM algorithm has an average rotation error of 19.588 degree in area B and 11.957 degree in area D. Such a large rotation error has a great impact on the mapping results, but the rotation error of CSM is still within a reasonable range in area A and area C. The rotation error of IMU-aided CSM method in B and D regions are much better than that in CSM, but compared with A, C region is still larger. Our algorithm is stable in the whole positioning process, and the rotation error is controlled within 1 degree, which greatly improves the positioning and mapping accuracy.

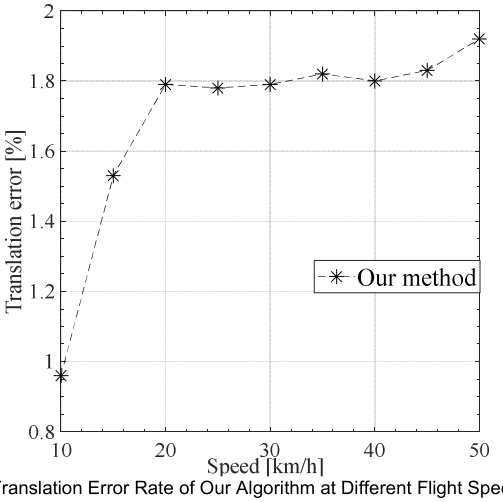
The CSM method is actually a brute-force matching algorithm. It needs to compare each pair of adjacent frames, which



(a) Translation Error Rate of CSM Algorithm at Different Flight Speeds

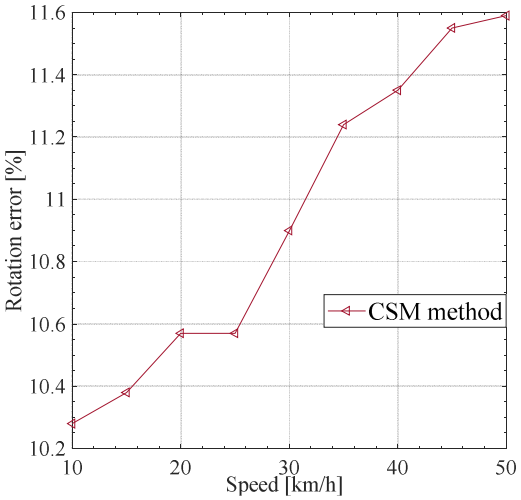


(b) Translation Error Rate of IMU-aided CSM Algorithm at Different Flight Speeds

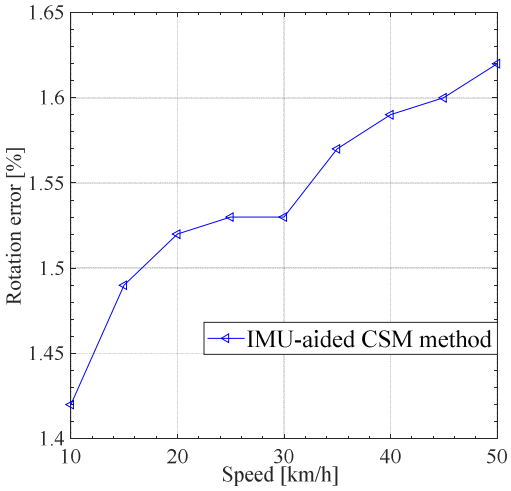


(c) Translation Error Rate of Our Algorithm at Different Flight Speeds

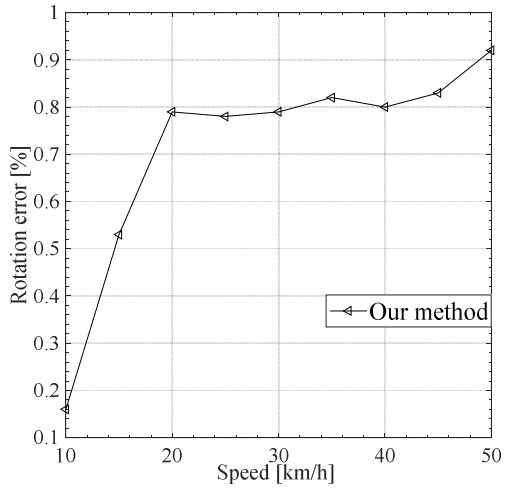
FIGURE 14. Comparison of translation errors of three algorithms at different flight speeds.



(a) Rotation Error Rate of CSM Algorithm at Different Flight Speeds



(b) Rotation Error Rate of IMU-aided CSM Algorithm at Different Flight Speeds



(c) Rotation Error Rate of Our Algorithm at Different Flight Speeds

FIGURE 15. Comparison of rotation errors of three algorithms at different flight speeds.

consumes more calculation time. Although the IMU used by the IMU-aided CSM method has a high frequency and a fast calculation speed, it still increases the calculation time to a certain extent. Our proposed algorithm uses secant slope matching at key frames, saving time for brute force matching

and improving accuracy. As shown in Table 4, t is the time of positioning and map update, and our algorithm can guarantee the real-time requirements of positioning.

D. EVALUATION OF ALGORITHMS AT DIFFERENT SPEEDS

We also use the flight speed (km / h) as the measurement unit to exclude the impact of the flight speed on the matching results. Including 10km / h, 15km / h, 20km / h, ...50km/h calculate their translation error percentage (%), and rotation error percentage (%). The results are shown in Figure 14(a), (b), (c) and figure 15(a), (b), (c).

It can be seen from Figure 14 and Figure 15 that both translation error rate and rotation error rate increase with the increase of flight speed. The growth speed is fast first, then slow, and finally tends to a certain range. The translation error rate of CSM algorithm is about 40%, that of IMU-aided CSM method is about 10%, and that of our method is about 2%. The rotation error rate of CSM algorithm is about 10%, that of IMU-aided CSM method is about 2%, and that of our method is within 1%. As can be seen above, the flight speed has a certain impact on the translation and rotation of the algorithm. The faster the speed is, the more serious the error is. However, the proposed method has far better positioning accuracy under fast flight conditions than the CSM algorithm and IMU-aided CSM algorithm.

VI. CONCLUSION AND DISCUSSION

In this paper, the influence of the lack of data in the transmission line map on the location algorithm is analyzed. The least square curve fitting method is used to parameterize the transmission line map, which enhances the geometric characteristics of the map model. At the same time, only the parameters of the fitting curve and the coordinates of each endpoint are stored for the transmission line, which greatly reduces the memory consumption. This paper analyzes the shortcomings of UAV pose estimation in the transmission line environment, proposes a matching method based on the secant slope of the transmission line, which reduces the three-dimensional geometric characteristics to one-dimensional slope characteristics, greatly simplifying the complexity of the algorithm; at the same time, based on the parabola-like characteristics of power lines, fast search and matching are performed using the dichotomy method. While reducing the translation error and rotation error of the drone, the efficiency of the algorithm is greatly improved. At last, the superiority of this algorithm is proved by comparing with CSM algorithm and IMU-aided CSM algorithm through several experiments. This article is a study of the power transmission line. In future work, we need to consider how to simplify the point cloud data in the tower map on the premise that the geometric characteristics of the obtained tower environmental information are not changed. Our algorithm is based on the theoretical situation of only two transmission lines between two power towers. In the future, we need to consider how to extract the features of the secant line in the case of multiple lines.

ACKNOWLEDGMENT

The authors would like to thank the editor and anonymous reviewers for their thoughtful comments and suggestions that have improved the quality of the article.

REFERENCES

[1] J. A. Cooney, W. L. Xu, and G. Bright, "Visual dead-reckoning for motion control of a mecanum-wheeled mobile robot," *Mechatronics*, vol. 14, no. 6, pp. 623–637, Jul. 2004.
[2] G. Krüger, R. Springer, and W. Lechner, "Global navigation satellite systems (GNSS)," *Comput. Electron. Agricult.*, vol. 11, no. 1, pp. 3–21, 2007.
[3] T.-S. Bae, D. Grejner-Brzezinska, G. Mader, and M. Dennis, "Robust analysis of network-based real-time kinematic for GNSS-derived heights," *Sensors*, vol. 15, no. 10, pp. 27215–27229, Oct. 2015.
[4] F. Pomerleau, F. Colas, and R. Siegwart, "A review of point cloud registration algorithms for mobile robotics," *Found. Trends Robot.*, vol. 4, no. 1, pp. 1–104, 2015.
[5] J. Andrade-Cetto and A. Sanfeliu, "The effects of partial observability in SLAM," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, 2004, pp. 397–402.
[6] S. J. Julier and J. K. Uhlmann, "New extension of the Kalman filter to nonlinear systems," *Proc. SPIE*, vol. 3068, pp. 182–193, Jul. 1999.
[7] G. Grisettiyz, C. Stachniss, and W. Burgard, "Improving grid-based SLAM with rao-blackwellized particle filters by adaptive proposals and selective resampling," in *Proc. IEEE Int. Conf. Robot. Autom.*, Apr. 2005, pp. 2432–2437.
[8] K. Konolige, G. Grisetti, R. Kümmerle, W. Burgard, B. Limketkai, and R. Vincent, "Efficient sparse pose adjustment for 2D mapping," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Oct. 2010, pp. 22–29.
[9] S. A. Hiremath, G. W. A. M. van der Heijden, F. K. van Evert, A. Stein, and C. J. F. ter Braak, "Laser range finder model for autonomous navigation of a robot in a maize field using a particle filter," *Comput. Electron. Agricult.*, vol. 100, pp. 41–50, Jan. 2014.
[10] C. Stachniss, "Particle filters for robot navigation," *Found. Trends Robot.*, vol. 3, no. 4, pp. 211–282, 2012.
[11] T. Bailey, J. Nieto, J. Guivant, M. Stevens, and E. Nebot, "Consistency of the EKF-SLAM algorithm," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Oct. 2006, pp. 3562–3568.
[12] J. A. Castellanos, J. Neira, and J. D. Tardós, "Limits to the consistency of EKF-based SLAM," *IFAC Proc. Volumes*, vol. 37, no. 8, pp. 716–721, Jul. 2004.
[13] J. Hong, L. Rui, S. Yingjing, and Y. Guo, "A series-wound EKF algorithm for attitude estimation," in *Proc. 29th Chin. Control Decis. Conf. (CCDC)*, May 2017, pp. 3847–3852.
[14] D.-J. Jwo and C.-H. Tseng, "GPS navigation processing using the IMM-based EKF," in *Proc. 3rd Int. Conf. Sens. Technol.*, Nov. 2008, pp. 589–594.
[15] B. Zhou, Y. Peng, and J. Han, "UKF based estimation and tracking control of nonholonomic mobile robots with slipping," in *Proc. IEEE Int. Conf. Robot. Biomimetics (ROBIO)*, Dec. 2007, pp. 2058–2063.
[16] M. Effati and K. Skonieczny, "EKF and UKF localization of a moving RF ground target using a flying vehicle," in *Proc. IEEE 30th Can. Conf. Electr. Comput. Eng. (CCECE)*, Apr. 2017, pp. 1–4.
[17] R. Tiar, M. Lakrouf, and O. Azouaoui, "Fast ICP-SLAM for a bi-steerable mobile robot in large environments," in *Proc. Int. Conf. Adv. Robot. (ICAR)*, Jul. 2015, pp. 1–6.
[18] S. Li, J. Wang, Z. Liang, and L. Su, "Tree point clouds registration using an improved ICP algorithm based on kd-tree," in *Proc. IEEE Int. Geosci. Remote Sens. Symp. (IGARSS)*, Jul. 2016, pp. 4545–4548.
[19] P. Biber and W. Strasser, "The normal distributions transform: A new approach to laser scan matching," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, vol. 3, Oct. 2003, pp. 2743–2748.
[20] C. Ulaş and H. Temeltaş, "3D multi-layered normal distribution transform for fast and long range scan matching," *J. Intell. Robotic Syst.*, vol. 71, no. 1, pp. 85–108, Sep. 2012.
[21] V. Fox, J. Hightower, L. Liao, D. Schulz, and G. Borriello, "Bayesian filtering for location estimation," *IEEE Pervas. Comput.*, vol. 2, no. 3, pp. 24–33, Sep. 2003.
[22] J. Biswas and M. Veloso, "Episodic non-Markov localization: Reasoning about short-term and long-term features," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2014, pp. 3969–3974.

[23] E. B. Olson, "Real-time correlative scan matching," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2009, pp. 4387–4393.
[24] E. Olson, "M3RSM: Many-to-many multi-resolution scan matching," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2015, pp. 5815–5821.
[25] W. Hess, D. Kohler, H. Rapp, and D. Andor, "Real-time loop closure in 2D LIDAR SLAM," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2016, pp. 1271–1278.
[26] M. Newville, T. Stensitzki, and D. B. Allen, *LMFIT: Non-linear least-square minimization and curve-fitting for Python*. Chicago, IL, USA: Astrophysics Source Code Library, 2016.
[27] D. Li, B. Zhang, and C. Li, "A feature-scaling-based k-nearest neighbor algorithm for indoor positioning systems," *IEEE Internet Things J.*, vol. 3, no. 4, pp. 590–597, Aug. 2016.
[28] P. R. C. Mendes, J. M. Maestre, C. Bordons, and J. E. Normey-Rico, "Binary search algorithm for mixed integer optimization: Application to energy management in a microgrid," in *Proc. Eur. Control Conf. (ECC)*, Jun. 2016, pp. 2620–2625.
[29] C. Biernacki and J. Jacques, "Model-based clustering of multivariate ordinal data relying on a stochastic binary search algorithm," *Statist. Comput.*, vol. 26, no. 5, pp. 929–943, Jun. 2015.
[30] Chang'an. Liu, Ziqiang. Cai, Changhao. Sun, "IMU and CSM fusion localization algorithm based on environmental assessment," *J. Huazhong Univ. Sci. Technol. Natural Sci. Ed.*, vol. 46, no. 12, pp. 1671–4512, 2018.
[31] R. Kümmerle, B. Steder, C. Dornhege, M. Ruhnke, G. Grisetti, C. Stachniss, and A. Kleiner, "On measuring the accuracy of SLAM algorithms," *Auto. Robots*, vol. 27, no. 4, pp. 387–407, Sep. 2009.

CHANG'AN LIU was born in Heilongjiang, China, in 1971. He received the Ph.D. degree in computer applied technology from the Harbin Institute of Technology, in 2001. From 2001 to 2019, he was engaged in teaching and scientific research at the School of Control and Computer Engineering, North China Electric Power University, Beijing, China. He was also the Director of Education, a Professor of the School of Control and Computer Engineering, and the Director of the Institute of Intelligent Robots. He is currently the Vice President of the North China University of Technology, Beijing, China. His research interests include intelligent robot technology, artificial intelligence and applications, embedded systems and applications, graphics, images, and information visualization technology.

YUE SHAO was born in Heilongjiang, China, in 1996. She received the B.S. degree in the Internet of Things engineering from Northeast Petroleum University, Daqing, in 2018. She is currently pursuing the master's degree in software engineering with the School of Control and Computer Engineering, North China Electric Power University, Beijing, China. Her research interests include artificial intelligence, power-grid technology, drone inspections, and machine learning.

ZIQIANG CAI was born in Jiangxi, China, in 1994. He received the bachelor's degree in automation from Northeastern Electric University, in 2016, and the master's degree in computer application technology from North China Electric Power University, Beijing, in 2019. His research direction interests include intelligent robot technology and power system drone inspection.

YUEJIE LI was born in Shandong, China, in 1978. She received the master's degree in computer application technology from Tianjin Polytechnic University, China, in 2007. She is currently pursuing the Ph.D. degree with North China Electric Power University. She is also an Associated Professor with the Department of Mathematics and Computer Engineering, Ordos Institute of Technology, China. Her research interests include artificial intelligence, pattern recognition, intelligence computation, and embedded systems.