Multi-sensor fusion localization algorithm for outdoor mobile robot

Chang Li1, Hanxu Sun2, Ping Ye3

1Beijing University of Posts and Telecommunications Beijing, China
2Beijing University of Posts and Telecommunications Beijing, China
3Beijing University of Posts and Telecommunications Beijing, China

Abstract. In this paper, the multi-sensor fusion positioning technology of outdoor mobile robot is studied, and a real-time outdoor positioning algorithm combining GPS and lidar odometry is proposed. Lidar odometry based on 3D lidar for point cloud feature matching is realized. At the same time, a covariance matrix reflecting the local positioning information of the lidar odometry is constructed in real time according to the matching distance and the absolute median difference, and the accuracy factor, the circular probability error, etc. are used as the standard GPS. The positioning effect is evaluated to measure the respective positioning effects of the lidar odometry and GPS. On this basis, the GPS is used as the global factor in the pose structure, and the lidar odometry is used as the local factor to construct the pose observation constraint, and the global nonlinear trajectory optimization is carried out to realize the outdoor real-time positioning of the mobile robot. Experimental results and data analysis verify the effectiveness and practicability of the proposed method.

1. Introduction
Outdoor positioning is one of the key technologies of intelligent mobile robots. With the rapid development of cleaning robots and distribution robots, it shows that outdoor positioning has great application prospects and research value.

Outdoor mobile robots, such as patrol robots and cleaning robots, are low-speed mobile platforms. The working environment includes both structured and unstructured roads, and there are many dynamic objects (vehicles, pedestrians). Therefore, the outdoor mobile robot is required to avoid collision in the working area, and high requirements are placed on the positioning accuracy, stability and robustness at work. The mobile robot can perform sensing positioning based on sensors such as cameras, lidars, and GPS. However, due to the size and constraints of the computing platform, it is impossible to perform calculations with a large amount of calculation. Outdoor mobile robots require real-time positioning in complex and varied scenes. How to stabilize and accurately position mobile robots in outdoor scenes becomes the subject of this paper.

The positioning in the outdoor scene is inseparable from GPS. The GPS-based positioning method can perform continuous positioning around the clock. The differential GPS can realize centimetre-level positioning and is suitable for global positioning, but it is affected by the environment, high buildings and trees, the tunnel, etc. will block the GPS signal [1]. The lidar odometry can provide precise pose [2] in a small area, which is not affected by illumination. But it will have a large cumulative error after long working hours [3]. In addition, GPS data can provide global road information for mobile robots and a priori data for path planning of mobile robots. It also can improve the robustness and security of path...
planning. The filter-based fusion method is sensitive to time synchronization, and any data delay measurement has problems, such as EKF [4], MSCKF [5]. The optimization-based fusion algorithm maintains a large number of measured values through global optimization and calculates the pose by one optimization. Compared with the filter method, the problem caused by time synchronization is relatively small, and the accuracy is higher under the same computational complexity such as VINS-MONO [6], VINS-fusion [7]. Based on the time synchronization and accuracy requirements of this paper, this paper uses graph optimization to perform multi-sensor pose fusion. Based on the positioning status and other information in the GPS, the matching distance and statistical information in the lidar odometry construct the covariance matrix to update the pose of the mobile robot, and use the optimized pose as the basis for the loop closure detection of the lidar odometry [8], further improving the accuracy of outdoor global positioning.

The organizational structure of this paper is as follows: The first section describes the positioning problem of mobile robots in practical applications in outdoor scenes. The second section introduces the multi-sensor fusion localization algorithm proposed in this paper. In the third part, this paper describes in detail the method of GPS and lidar odometry pose fusion based on pose optimization. The experiments conducted and the results are shown in the fourth quarter, and the subsequent conclusions are in the fifth section.

2. System overview

The multi-sensor fusion positioning algorithm in this paper is mainly divided into three parts. The overall fusion positioning system is shown in figure 1. This article uses a sliding window to optimize the fusion of GPS data and lidar odometries. Considering GPS global drift-free characteristics, this paper uses GPS as global factor. Considering the local precise characteristics of the lidar odometry, the lidar odometry is used as local factor to construct a pose map. The lidar odometry uses a multi-line lidar as a sensor to perform pose estimation based on the feature matching of the three-dimensional point cloud, and performs light weight processing to reduce the amount of calculation on the mobile platform.

![Figure 1. Fusion positioning system](image-url)
The second part is the construction of the covariance matrix. In statistics, the covariance matrix is used to characterize the correlation between random variables, reflecting the statistical properties between variables. In multi-sensor localization fusion, the covariance matrix is usually used to update the proportion of the pose at the time of fusion. This paper creates a covariance matrix reflecting the current GPS positioning state based on the accuracy factor and other information in the GPS data, and creates a lidar odometry covariance matrix based on the results of the lidar current point cloud and point cloud map matching. The fusion algorithm outputs a stable and accurate pose by adding the covariance matrix of the GPS and lidar odometry to the fusion algorithm for real-time updating.

The third part is loop closure detection. As a pose estimation, lidar odometry has very high precision in short-distance local positioning, but it will produce large cumulative error in long-term positioning. By using the combined pose as the basis for the loop closure detection, the cumulative error of the lidar odometry can be greatly reduced, and the accuracy of the post-fusion positioning can be further improved.

3. Optimized methods

3.1. Construction multi-sensor fusion system

In order to ensure that the fusion algorithm can stably output the pose based on the higher frequency, this paper uses the sliding window to optimize the fusion of GPS data and lidar odometry, using GPS as the global factor, and using the lidar odometry as a local factor to construct the pose graph.

The pose graph structure of the fusion algorithm is shown in figure 2:

Figure 2. An illustration of the pose graph structure. The nodes (circles) in the pose graph are each pose in the world pose coordinate system, and the frequency is determined by the frequency of the lidar odometry. The edges between two consecutive nodes are locally constrained by lidar odometry; the other edges are globally constrained by GPS.

Pose graph optimization is a maximum likelihood estimation problem. If the measurement is independent and obeys the Gaussian distribution, it can be transformed into a nonlinear least squares problem. The local factor is constructed by the relative pose between two consecutive frames obtained by the lidar odometry, and the global factor is constructed by GPS data. Since the general GPS data only provides the position information, only the position information is added when the global factor is created. When using GPS as a factor, its measurements directly constrain the position of each node. The covariance is determined by the state of receiving satellite signals and the other conditions.

After adding all the factors, the pose graph optimization is performed as follows:

- when there is a new GPS signal reception, the data fusion is going to start;
- Using the cere library for optimization, the parameters for adding the optimization variables are determined by the lidar odometry, that is, the seven variables of the position and orientation of the robot \( \{ p_x, p_y, p_z, q_x, q_y, q_z \} \):
- Construct residual. The GPS factor is the absolute error and the lidar odometry factor is the relative error. Among the observations, the initial position is provided by GPS, and the lidar odometry observation trusts the pose change of the two frames of data;
- Add the covariance matrix. The covariance precision is taken as the denominator of the optimization. The higher the precision, the greater the weight of the influence on the pose;
3.2. Construct pose covariance matrix

3.2.1. Construct GPS covariance.

The GPS positioning system uses the basic triangulation principle of the satellite and the GPS receiving device to measure the transmission time of the radio signal to measure the distance. Differential GPS uses differential technology to eliminate or reduce errors caused by satellite clocks and broadcast delays, allowing GPS to achieve higher accuracy. The current high-precision military differential GPS can achieve centimeter-level accuracy in both static and ideal environments. The ideal environment here means that there is not too much suspended medium in the atmosphere, and GPS has a strong receiving signal when measuring. However, mobile robots are driven in complex dynamic environments, especially in large cities where GPS multipath reflections are more pronounced. The GPS positioning information thus obtained is easily subject to a few meters of error, which is likely to cause an accident.

Therefore, the GPS positioning covariance is constructed in combination with the satellite positioning state, the positioning factor, and the positioning error to describe the current positioning state. The common output data format of GPS is the GPGGA statement, which is one of the most widely used data in the NEMA format. The statement contains information such as longitude, latitude, altitude, location quality, number of satellites used, and horizontal longitude factor. In addition, the use of different GPS receiving devices also has different positioning errors. Combine all the above information to construct the GPS positioning covariance and build a model:

- Set the circular probability error. The positioning accuracy unit of different GPS devices is determined by the circular probability error CEP, which refers to the specified error with 50% probability. In this paper, we use the specified error at 95% probability. For the GPS device used in this paper, the error is 1.5m/2.5m (horizontal/altitude) when the single point is positioned, 40cm/80cm is the positioning error when the DGNSS is positioned, and the fixed de-positioning error is the 1cm/1.5cm when the RTK is positioned. The floating point solution positioning error is 10cm/15cm;
- Get HDOP (horizontal position accuracy factor). The HDOP reflects the positioning quality of the GPS positioning in the horizontal position. The smaller the HDOP value, the higher the positioning accuracy;
- Construct covariance matrix. According to the received GPS statement, the current positioning state is obtained, the corresponding positioning error is selected, and then the variance is calculated according to the horizontal positioning factor, and the diagonal element of the covariance matrix is constructed. The specific construction process is as follows:

\[
\begin{align*}
COV_\text{X} &= (HDOP \times \text{lon } \text{err})^2 \quad \text{longitude} \\
COV_\text{Y} &= (HDOP \times \text{lat } \text{err})^2 \quad \text{latitude} \\
COV_\text{Z} &= (HDOP \times \text{alt } \text{err})^2 \quad \text{altitude}
\end{align*}
\]

Where \( COV_\text{X} \), \( COV_\text{Y} \), and \( COV_\text{Z} \) are the variances in the longitude, latitude, and altitude directions, and \( \text{lon } \text{err} \), \( \text{lat } \text{err} \), and \( \text{alt } \text{err} \) are the circular probability errors. The latitude of the entire covariance matrix is 3\( \times \)3, the diagonal element is the variance calculated by the above formula, and the remaining elements are filled with the maximum value.

3.2.2. Construct lidar odometry covariance matrix.

The lidar odometry used in this paper creates a 3D point cloud map while providing the pose of the robot. The ICP matching can completely overlap the two point cloud coincident parts by translating and rotating the point cloud. Therefore, the accuracy of the positioning can be described based on the ICP matching of the point cloud (the iterative closest point), and the lidar odometry covariance matrix is created.
Perform ICP matching based on the current frame point cloud and point cloud map. According to the result of ICP matching, the closest point distance error of the two sets of point cloud matching can be obtained;

According to the distance error calculated in (1), the median absolute deviation is used to measure the quality of the point cloud matching, and the covariance matrix is constructed;

In statistics, median absolute deviation (MAD) is a robust measure of the sample bias for univariate numerical data. It can also represent the overall parameters derived from the MAD of the sample. MAD is a robust statistic that is more adaptive to outliers in the dataset than standard deviation. For the standard deviation, the square of the distance from the data to the mean is used, so the large deviation weight is larger, and the outliers also have an important effect on the result. For MAD, a small amount of outliers will not affect the final result. For univariate data \( X_1, X_2, \ldots, X_n \), MAD is defined as the median of the absolute deviation of the data point to the median:

\[
\text{MAD} = \text{median}(|X_i - \text{median}(X)|)
\]

That is, the residual between the data and their median is calculated first, and the MAD is the median of the absolute values of these residuals. The closest point distance obtained by ICP matching in this paper is a distance error. In order to obtain a suitable absolute median difference, the original median needs to be added to the absolute median difference. On the basis of obtaining the absolute median difference, it is necessary to associate the absolute median difference with the standard deviation to obtain the final covariance matrix. Therefore, in order to use MAD as a consistent estimator of the standard deviation \( \sigma \) estimate, the following formula is used:

\[
\sigma = k \cdot \text{MAD}
\]

Where \( k \) is the scale factor constant, for the 1/4 to 3/4 interval of the standard normal distribution,

\[
k = 1\left(\Phi^{-1}(3/4)\right) \approx 1.4826, \text{ that is } \sigma=1.4826 \cdot \text{MAD}
\]

Thus, lidar odometry covariance matrix based on ICP matching can be created based on the standard deviation.

3.3. loop closure detection

Since the lidar odometry only considers the key frames in adjacent time, the previously generated errors will accumulate to the next moment. As the scene expands, the long-term estimated pose will produce a large cumulative error, making it impossible to construct globally consistent trajectories and maps. Therefore, according to the fusion results of the previous chapter, the lidar odometry is provided with a more reliable loop closure detection judgment basis, and the entire map is optimized according to the point cloud matching method.

In order to improve the computational efficiency of the loop closure detection and optimization of the lidar odometry, a separate point cloud map will not be created, but the corresponding feature points will be saved as pose constraints. According to the merged pose, when the mobile robot detects the return to the vicinity of the previous position, the loop closure detection is performed, and the entire pose graph is updated according to the result of the point cloud matching. The specific process is:

- Extract key frames and save feature points. Since the optimization speed is slow, the key frames (scheduled sampling track points) are extracted for matching;
- Judge loop closure detection. When it is judged that the mobile robot returns to the vicinity of the previous position according to the merged pose, the radius search is performed to obtain a nearby track point. If it is determined that a key frame in the vicinity is equal to a previous key frame, it is considered that a loop closure is detected, and matching and optimization are started;
- matching optimization. After the loop closure is detected, the current point cloud is superimposed and ICP matched with the detected point cloud at the key frame. Then, based on the result of the matching, the entire pose is updated and the point cloud is superimposed;

At this point, a stable pose output is achieved based on the fusion algorithm of GPS and lidar odometry.
4. Experimental result

In order to verify that the proposed algorithm can solve the problems faced by mobile robots in outdoor positioning, the proposed algorithm is used to test the mobile robot platform. It is necessary to use the fusion positioning algorithm experimental platform to ensure that the mobile robot performs stable navigation and path planning in the outdoor scene. The experimental platform uses the riki-robot mobile robot developed by the laboratory. The computing platform uses the Intel core i5-200 @ 2.20hz processor. The frequencies of GPS and Lidar are 5hz and 10hz respectively.

4.1. Global path test

We input the lidar odometry and GPS data as pose graph where the GPS data contains the covariance matrix of position estimation, and the icp-match can obtain the covariance matrix of the match which can be used when the global path is updated.

![Figure 3. Global path comparison](image)

As shown in figure 3, our lidar odometry has a large cumulative error while our merged path almost coincides with the true value. As can be seen from the trajectory, the intermediate process of lidar odometry has a short static drift, but due to the presence of GPS, the merged path is still very smooth.

4.2. Global position test

Due to the actual scene, the positional deviation of the robot is different, the error in the x-axis direction is large, and the error in the y direction is small. In addition, occlusion in different directions can also cause errors in GPS data in different directions.

![Figure 4. Global position comparison](image)

As shown in figure 4, we plot the difference of the fusion position and true value. Since the robot moves on a two-dimensional plane, the z-axis deviation is small, so it is ignored here. It can be seen from the figure that the deviation of the x-axis direction lidar odometry is large, and the GPS data has a certain fluctuation in the y-axis direction.

5. Conclusive

The multi-sensor fusion algorithm of outdoor mobile robot proposed in this paper combines the characteristics of GPS and lidar odometry to ensure stable and accurate positioning of mobile robots in complex outdoor scenes. The use of pose optimization method ensures the fusion of GPS and lidar odometry at lower frequencies, which improves the robustness of the algorithm. The update of each part of the covariance matrix ensures that the system can perform effective pose fusion according to the current scene location, which ensures the accuracy of the pose of the mobile robot after fusion. At the same time, the lidar odometry can provide a more accurate pose after adding loop closure detection, ensuring the global consistency of mobile robot positioning and mapping. The experimental results show...
that the proposed algorithm can effectively provide a stable and accurate pose for outdoor mobile robots, ensuring that it can complete outdoor operations for navigation and has high engineering applicability.

References

[1] Sośnica K, Bury G, Zajdel R, et al. Estimating global geodetic parameters using SLR observations to Galileo, GLONASS, BeiDou, GPS, and QZSS[J]. Earth Planets and Space, 2019, 71(20).

[2] Renaud Dubé, Dugas D, Stumm E, et al. SegMatch: Segment based place recognition in 3D point clouds[C]// 2017 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2017.

[3] Zhang J, Singh S. Low-drift and real-time lidar odometry and mapping[J]. Autonomous Robots, 2017, 41(2):401-416.

[4] Heo S, Chan G P. Consistent EKF-based visual-inertial odometry on matrix Lie group[J]. IEEE Sensors Journal, 2018, PP(99):1-1.

[5] Sun K, Mohta K, Pfrommer B, et al. Robust Stereo Visual Inertial Odometry for Fast Autonomous Flight[J]. IEEE Robotics and Automation Letters, 2018, 3(2):965-972.

[6] Tong Q, Peiliang L, Shaojie S. VINS-Mono: A Robust and Versatile Monocular Visual-Inertial State Estimator[J]. IEEE Transactions on Robotics, 2018:1-17.

[7] Qin T, Cao S, Pan J, et al. A General Optimization-based Framework for Global Pose Estimation with Multiple Sensors[J]. 2019.

[8] Haoyang Ye, Yuying Chen, Ming Liu. Tightly Coupled 3D Lidar Inertial Odometry and Mapping[C]// 2019 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2019.