


Article

# Research on Map-SLAM Fusion Localization Algorithm for Unmanned Vehicle

Shuguang Li <sup>1,\*</sup> , Zhenxu Li <sup>1</sup>, Xinxin Liu <sup>2</sup>, Chunxiang Shan <sup>1</sup>, Yang Zhao <sup>1</sup> and Hong Cheng <sup>1</sup>

<sup>1</sup> School of Automation Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China

<sup>2</sup> Xi'an Modern Control Technology Research Institute, Xi'an 710065, China

\* Correspondence: lisg@uestc.edu.cn

**Abstract:** Vision-based localization techniques and detection technologies are key algorithms for the localization and navigation of unmanned vehicles. Especially in scenarios where GPS signals are missing, Simultaneous Localization and Mapping (SLAM) techniques that rely on vision, inertial navigation system (INS) and other sensors have important applications. Among them, vision combined with the IMU SLAM system has the advantage of realistic scale, which is lacking in monocular vision and computational power compared to multi-visual vision, so it is suitable for application in an unmanned vehicle system. In this paper, we propose a fusion localization algorithm that combines a visual-inertial SLAM system and map road information, processing road information in a map under structured roads, and detecting lane lines and locating its local position by a monocular camera, applying a strategy of position prediction and update for map-SLAM fusion localization. It solves the problem of accumulating errors in a pure SLAM system without loopback and provides accurate global-local positioning results for unmanned vehicle positioning and navigation.

**Keywords:** unmanned vehicle; SLAM; visual INS fusion; map localization



**Citation:** Li, S.; Li, Z.; Liu, X.; Shan, C.; Zhao, Y.; Cheng, H. Research on Map-SLAM Fusion Localization Algorithm for Unmanned Vehicle. *Appl. Sci.* **2022**, *12*, 8670. <https://doi.org/10.3390/app12178670>

Academic Editors: Andrea Prati and Rocco Furferi

Received: 12 June 2022

Accepted: 22 August 2022

Published: 29 August 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

In the positioning requirements of unmanned driving, the global positioning system based on GPS and differential GPS has the characteristics of high efficiency and accuracy. However, in tunnels, heavily blocked roads, urban canyons and other roads, GPS signals are severely affected and cannot complete the positioning work. At this time, SLAM technology has important application value [1,2]. Compared with other sensors, vision technology can better combine perceptual information and play a major role in unmanned vehicle applications. Vision sensors can detect lane lines [3,4], road signs [5,6] and other information required for vehicle driving through deep learning and other technologies. Under structured roads, good detection results can be achieved for unmanned driving needs, through accurate local positioning methods, such as lane keeping [7,8], navigable area navigation [9], etc. However, this local navigation method still needs SLAM and other methods to supplement and judge the positioning results due to the instability of detection. The visual SLAM system mainly includes two forms: monocular [10–13] and multi-eye [14]. Multi-eye vision requires complex calculations and a large amount of matching point information. Monocular vision is simple and stable, but it lacks the true scale. The SLAM system [15,16] combined with a monocular camera and IMU has better adaptability, and it can provide short-term stable positioning results in scenarios where GPS is missing. However, in the application of unmanned vehicles, the positioning result error will continue to increase due to accumulated errors and IMU divergence, and the trajectory of unmanned vehicles can rarely be corrected by loopback [17].

To address the problem of accumulating errors in SLAM for unmanned vehicles, we introduce map-prior information for SLAM repositioning. The accumulated visual errors

and IMU integration lead to increasing SLAM localization errors, while the single movement of an unmanned vehicle rarely passes through repeated positions and cannot be corrected by traditional loopback. We adopt a maximum a posteriori probability approach to represent its own position through a probability distribution; use the localization result output of SLAM as the position update incentive; use the map midway width, lane angle and road position information as a priori; and adopt a prediction-update model to calculate the probability distribution of vehicle position through the recognized lane line perception information. It is worth noting that, different from the general map trajectory mapping [18] which compares a certain segment of motion results with map information, the semantic SLAM [19,20] method increases the positioning accuracy by adding the inter-frame connection of semantic information. The algorithm flow is shown in Figure 1.

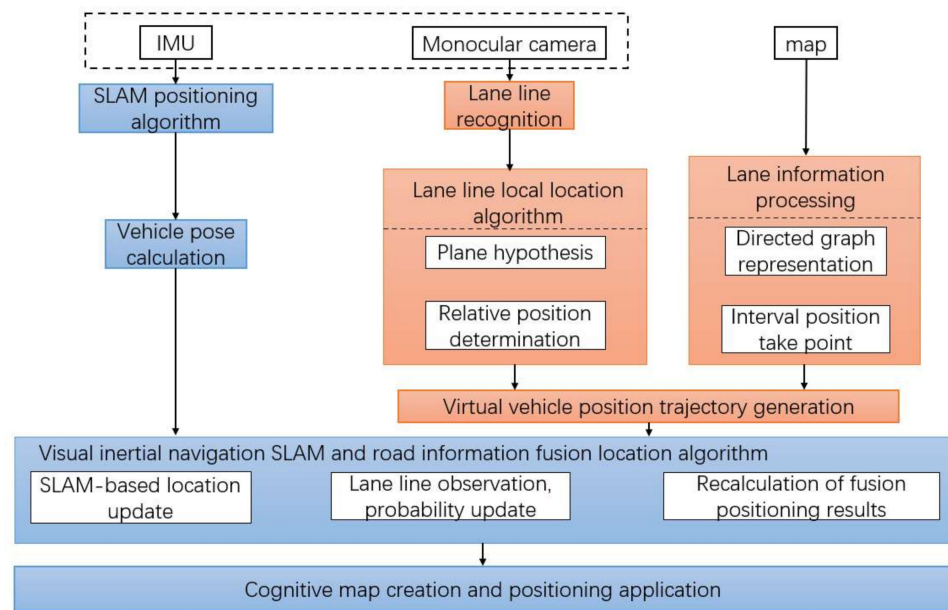


Figure 1. Algorithm flow diagram.

## 2. Structured Representation of Lane Lines and Map Information

### 2.1. Lane Line Detection and Positioning

To combine the detected lane information with the lane information in the map, it is necessary to structure the lane lines and the map information separately. In this paper, lane line detection is performed by [3], and the split-shuffle-bottleneck [21] mechanism is introduced to improve the detection speed. At the same time, a fixed-parameter perspective transformation matrix is used instead of the adaptive H-Net [22], which further improves the speed and accuracy.

The obtained lane line detection result is a point set on the image plane. By projecting the point set to 3D homogeneous coordinates  $(x, y, 1)$ , a straight line is fitted to obtain the lane line  $l_1l_2..l_n$ , and the normal vector  $d$  between the lane line. The center is calculated, and the direction can be judged by the relative position of the exit lane line and the vehicle. The left and right lane lines  $l_{left}l_{right}$  are selected by the two nearest neighbors.

To be able to combine with the data in the map, some key information corresponding to the two needs to be obtained. We make the road plane assumption, which is that the points of the lane line  $l_{left}l_{right}$  are in this plane. It means that the road is assumed to be a plane, or the vehicle. The short driving area is a flat surface, and the car runs smoothly on this road surface without leaning forward or backward at a large angle.

Suppose that in the camera coordinate system, the road plane equation  $plane_A$  is:

$$Ax + By + Cz + D = 0 \tag{1}$$

According to the road plane assumption, when the vehicle is running smoothly, the relative position relationship between the camera and the road plane will not change, so the plane parameter  $ABCD$  can be obtained by calibration.

$$\begin{cases} f_x \cdot \frac{X_C}{Z_C} + u_0 = u \\ f_y \cdot \frac{Y_C}{Z_C} + v_0 = v \\ A \cdot X_C + B \cdot Y_C + C \cdot Z_C + D = 0 \end{cases} \quad (2)$$

Among them,  $f_x, f_y, u_0, v_0$  represent the pixel focal length and center offset in the camera internal parameters, and  $s$  represents the position scale of the monocular camera. The equations  $u, v$  and the camera internal parameters are all known, so only four sets of points are needed.  $X_C, Y_C, Z_C, X_C, Y_C, Z_C$  can calculate the plane parameters, that is, the plane parameter results can be obtained by calibrating the known points a priori. After the plane equation is known, to obtain the true three-dimensional image lane line point  $P_0 = [X_C, Y_C, Z_C]^T$ , the description of  $P_0$  observed by the camera is:

$$p_0 = s \cdot K \cdot P_0 \quad (3)$$

$$K = \begin{bmatrix} f_x & 0 & u_0 \\ 0 & f_y & v_0 \\ 0 & 0 & 1 \end{bmatrix} \quad (4)$$

$$p_0 = [u, v, 1]^T \quad (5)$$

Simplify to obtain camera constraints:

$$\begin{cases} f_x \cdot \frac{X_C}{Z_C} + u_0 = u \\ f_y \cdot \frac{Y_C}{Z_C} + v_0 = v \end{cases} \quad (6)$$

Therefore, using the equation again, we can solve the true value of  $P_0 = [X_C, Y_C, Z_C]^T$ , and finally obtain the positioning result of each point of the lane line, so as to calculate the equation of  $l_{left}l_{right}$  in the camera coordinate system:

$$l : \begin{cases} AX + BY + CZ + D = 0 \\ A_1X + B_1Y + C_1Z + D_1 = 0 \end{cases} \quad (7)$$

Among them, the two planes are, respectively, denoted as the road surface plane  $plane_A$  and the plane perpendicular to the road surface  $plane_B$ , which are obtained by the lane line 3D point coordinates  $P_0 = [X_C, Y_C, Z_C]^T$ . Therefore, the relative position relationship between the lane line and the vehicle can be calculated, including the absolute distance  $d_{lane} = d(plane_B, (0, 0, 0))$  of the vehicle from the left and right lane lines, and the projection angle of the lane line:

$$\theta_{lane} = \arcsin\left(\frac{d_{lane}}{d((0, 0, 0), P_0)}\right) \quad (8)$$

$$P_0 = \begin{cases} A_1X + B_1Y + C_1Z + D_1 = 0 \\ l_0 = k^*(0, 0, 1) \end{cases} \quad (9)$$

## 2.2. Map Information Processing

Usually, the latitude and longitude of a road centerline, the width of the road, and the slope of the road can be obtained on the map, but this result is not easy to use directly, so this article first processes the data by straight curve fitting. Use the outlier elimination strategy based on Random SAmple Consensus(RANSAC) [23] to represent the longitude and latitude of the lane centerline to fit the representation, fit the two lane lines through

the lane centerline and the road width, and calculate the structured representation of the two-lane lines, respectively.

As show in Figures 2 and 3. This paper adds road width information  $d$  based on road information, obtains the lane line information after translation, translates the distance between the lane and the lane line to the left and right, and obtains the left lane line  $l_{left}$  and the right lane line  $l_{right}$ , as well as performs the position-preserving sequence KD-Tree, which is used for storage and is more convenient for a subsequent search.

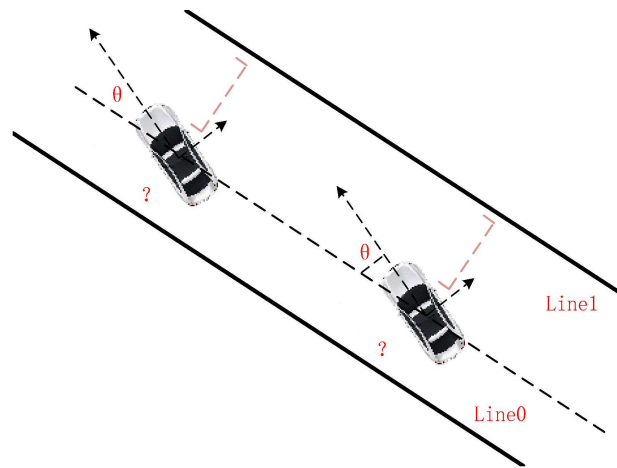


Figure 2. Representing map information that is difficult to use directly through directed graphs.

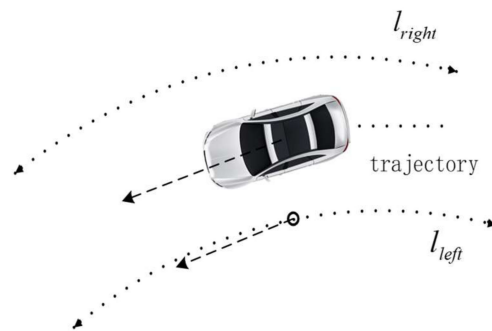


Figure 3. Schematic diagram of lane virtual candidate points and feature matching.

### 3. Fusion of Lane Location Information and SLAM Positioning Algorithm

The above-mentioned lane line detection and representation are sufficient to enable unmanned vehicles to complete lane keeping and steering functions, but it is still impossible to determine the specific global position of the lane where the unmanned vehicle is located. Therefore, the global consistent positioning method of SLAM needs to be considered. This paper uses a SLAM system that combines vision and inertial navigation. Based on the algorithm [15], IMU and monocular camera are used to complete the front-end output process of SLAM. Different from the indoor SLAM algorithm, it is difficult to have enough loop information during the driving process of the unmanned vehicle. Therefore, combining the timely position information which is provided by the SLAM and the map information is an important way to create an artificial loop. In this paper, by correlating the detected lane and position information, a prediction-update model is used to realize the fusion of SLAM positioning information and vehicle map lane position.

At each moment of fusion positioning, the result of the above local positioning  $d_{lane}$  is used to obtain the virtual trajectory of the vehicle parallel to the lanes of the map. Each point stored in the directed graph of the map is translated to this virtual trajectory in the same way. After the line localization result is updated, this operation performs the same processing on the virtual trajectory and the map point.

In the rest of this section, the map point mentioned refers to the point information stored in the directed graph on the map, and the candidate point refers to the map point where the vehicle has a greater probability of being at this position. The candidate points are on the virtual track obtained after each update of the distance from the lane line. The core of this method lies in the update of candidate points. Probability is used to update the candidate points and realize the positioning function.

### 3.1. Initialization of Starting Point Position

The result of SLAM output is the position in the body coordinate system, and the road information stored on the map is the global latitude and longitude. Therefore, first obtain the latitude and longitude coordinates and heading angle at the moment through manual positioning and location recognition at the starting point. The SLAM output is converted to latitude and longitude coordinates.

First, take the current position as the center point, and take the nearest  $K$  points in the map stored in the directed graph along the current yaw angle  $\theta$  a candidate points for the vehicle's possible poses. Each time a new candidate point is obtained, the probability of the vehicle at  $K$  candidate points on the map will be calculated, and the candidate point with the highest probability will be output as the vehicle position. At the initial moment, it is assumed that the probability of the vehicle at each candidate point of the map obeys the Gaussian distribution and satisfies the following formula.

$$\begin{aligned}
 P_{X_n} &= \lambda * \exp\left(-\frac{1}{2} * X^T * Q * X\right) \\
 X &= \{x - x_0, y - y_0\} \\
 \lambda &= \frac{1}{2\pi} |Q|^{-\frac{1}{2}} \\
 \sum_{n=1}^K P_{X_n} &= 1
 \end{aligned} \tag{10}$$

$P_{x_n}$  represents the probability that the vehicle position is at the current candidate point  $n$ ,  $Q$  represents the covariance matrix, which is set to the identity matrix at the initial moment; and  $X$  is the state variable at this time. At the initial moment, it only contains the position  $(x, y)$  in the world coordinate system;  $(x_0, y_0)$  is the map position obtained by initial positioning, that is, the center point of the Gaussian distribution; and the dimension of  $X$  is 2. At the same time, in order to avoid the influence of extreme conditions, after each probability calculation is completed, the probability of  $K$  points is normalized.

### 3.2. Location Prediction Based on SLAM Information

As shown in Figure 4. Firstly,  $d_{lane}$  is still obtained through the local positioning method of the lane line. The map candidate points are translated to the current vehicle position to generate a virtual trajectory, and a batch of new map candidate points are obtained. Taking the motion information from SLAM as the predicted excitation  $X_{k+1} = X_k + (\delta x, \delta y)$ ,  $(\delta x, \delta y)$  is the representation of the SLAM position change in the world coordinate system. After obtaining the changes combined with the SLAM positioning information, the position is updated on the basis of the  $K$  map candidate point positions obtained after the last calculation. For each candidate point, with the updated position  $X_{k+1} = X_k + (\delta x, \delta y)$  as the center, select  $K$  points according to the nearest neighbor principle, and calculate the probability of the updated position of the vehicle at each candidate point.

$$\begin{aligned}
 P_{X_k n}^{k+1} &= P_{X_k n}^{k+1} + P_{X_k n}^k * \lambda * \exp\left(-\frac{1}{2} X_{k+1}^T * Q * X_{k+1}\right) \\
 X_{k+1} &= \{x - x_{k+1}, y - y_{k+1}\} \\
 \sum_{n=1}^K P_{X_k n}^{k+1} &= 1
 \end{aligned} \tag{11}$$

$P_{X_k n}^{k+1}$  represents the probability that the vehicle is at position  $n$  at time  $k + 1$ . For each candidate point at time  $k$ ,  $K$  new candidate points are obtained and normalized. Through the prediction of  $K$  candidate points,  $K^2$  probability updates can be generated, and  $K_n$  new

candidate points  $K < K_n < K^2$  can be obtained. Through  $K$  calculations, the probability of the vehicle at each new candidate point is accumulated and normalized. Each candidate point has direction and position information in the representation of the directed graph, and can be established by constraining  $d_{lane}$  by the position of the lane line, the angle  $\theta_{lane}$  between the observed lane projection, the vehicle coordinate system and the map point contact, and update the probability of these  $K^2$  results as observations.

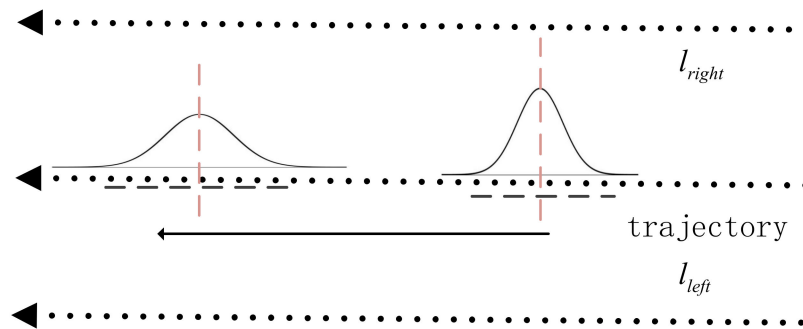


Figure 4. Probabilistic update diagram.

### 3.3. Location Probability Update Based on Lane Line Observation

As shown in Figure 5. This link needs to select the  $K$  points with the highest probability among the predicted candidate points. Since the lane of the map has been discretized by these points, first find the only corresponding lane line point  $map_n$  in the tangential direction of the vehicle attitude. This point is in the lane line stored in the original directed graph, representing the projection of the vehicle’s position on the lane line map point at this time. Observation is carried out at each candidate point, and the observed value is the distance  $d_{lane}$ , the angle of the lane line  $\theta_{lane}$  and the position of the vehicle body, which is obtained by the local positioning method of the lane line.

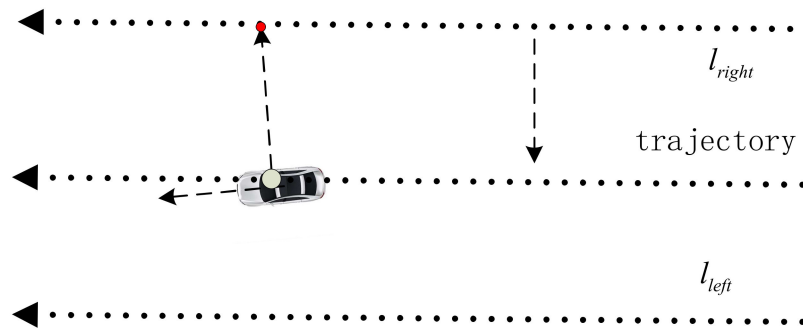


Figure 5. Recalculation of positioning information combined with map information.

The points in the map save the position information  $(x, y)$  and tangential angle information  $\theta_{kn}$  of each point, which can be combined with  $d_{lane}$  and angle  $\theta_{lane}$ . According to the current position of the vehicle, the distance to the map point in the direction of the tangential lane line can be obtained  $d_{kn} = \sqrt{(x_k - map_{nx})^2 + (y_k - map_{ny})^2}$ ,  $map_n$  represents the currently calculated map candidate point.  $\theta_{lane}$  represents the angle between the vehicle and the lane line. Set the current yaw angle of the vehicle as that which is output by the SLAM system and converted into world coordinates in the initialization process. At this time, the calculated lane angle is  $\theta_{mapn} = \theta_{lane} + \theta_k$ . It is worth noting that the lane line detection section above has fitted the lane line to the lane line closest to the current vehicle, so the probability of the vehicle at the current candidate point can be described by the probabilistic equation of the above parameters.

$$\begin{aligned}
 P_{X_{k+1}^n}^{k+1} &= P_{X_k^n}^{k+1} * \lambda * \exp\left(-\frac{1}{2} Y^T * D * Y\right) \\
 Y &= \{x - x_{k+1}, y - y_{k+1}, d_{kn} - d_{lane}, \theta_{kn} - \theta_{mapn}\} \\
 \sum_{n=1}^K P_{X_{k+1}^n}^{k+1} &= 1 \\
 \lambda &= \left(\frac{1}{2\pi}\right)^2 |D|^{-1/2}
 \end{aligned}
 \tag{12}$$

$P_{X_{k+1}^n}^{k+1}$  represents the probability of the latest position candidate point obtained by the vehicle after the above prediction, which is a correct update of  $P_{X_k^n}^{k+1}$ . The observations  $Y$  used to describe the vehicle at this candidate point include the latitude and longitude position  $(x, y)$ , the map distance point from the vehicle to the lane line, and the current. The error of the local positioning result  $(d_{kn} - d_{lane})$ , as well as the error of the angle  $(\theta_{kn} - \theta_{mapn})$  stored by the candidate point and the current observation angle, are a total of four-dimensional variables. The matrix  $D$  represents the covariance matrix. This paper assumes that each variable is distributed independently.  $D$  is a diagonal matrix with a dimension of  $4 \times 4$ , and the four diagonal corners  $(d_x, d_y, d_\theta, d_l)$  are, respectively, from the upper left to the lower right corner.  $(d_x, d_y)$  is the position variance,  $d_\theta$  is the angle variance with In radians and  $d_l$  represents the error of the current local positioning. Due to abnormal conditions such as detection failure, a robust kernel function  $h(x)$  is added to prevent the unstable influence caused by the wrong local positioning, and it is eliminated when the error is too large. Finally, all the candidate point probabilities are updated.

$$D = \begin{vmatrix} d_x, 0, 0, 0 \\ 0, d_y, 0, 0 \\ 0, 0, d_\theta, 0 \\ 0, 0, 0, d_l \end{vmatrix}
 \tag{13}$$

In this paper, only one lane line is used as the observation. In the experiment, the accuracy improvement produced by the use of multiple lane lines is very small, and at the same time, it generates a lot of problems, so that the data association difficult to determine. Therefore, the above operation only performs the observation update of one lane line. This paper takes the nearest left end lane line as an example. After the probability update is obtained, the  $K$  points with the highest probability are selected from the predicted candidate points  $K_n$  as the current new candidate points and normalized.

### 3.4. Result Processing and Output

After each observation update, the value with the highest probability  $K_{\max x} = \max_n (P_{X_{k+1}^n}^{k+1})$  is output as the current vehicle position relocation result. At the same time, according to the position  $K_{\max x}$  and direction angle of  $K_{\max \theta}$ , calculate the position of the current point and the original lane line point in the map, so as to obtain the updated local positioning result  $d_{lane}$ . Take the latest result obtained as the latest value and repeat the above steps to update the position.

The above results probabilistically process the position of the vehicle on the map. However, there are still some special problems that cause the system to fail to run, or make the convergence speed too slow:

a. There are not enough candidate points, or the probability value of the candidate points is too small.

When the system moves in a straight line for a long time, because the accumulated error is already large, the selected points are often limited to a certain range, resulting in the inability to update the position. The solution is that when the probability of a candidate point is too small, delete this point and let  $D' = 2 * D, Q = 2 * Q$  expand the value of the covariance matrix to facilitate the selection of more points next time.

b. Lack of significant optimal points.

If there is no significant optimal point  $P_{x_{\max}} > 0.3$  and  $P_{x_{\max}} > \sum_{x \in K} P_x$ , the probability difference of each candidate point is small at this time, and it is easy to fall into the local optimal value and cannot move forward. In places with obvious detours (intersections exceeding 90 degrees), the change of angle information is too obvious. Two candidate points with very different positions are likely to have similar probabilities. Therefore, at this time, it is necessary to randomly select candidate points, so that the predicted candidate points can be dispersed into different possibilities as much as possible, and the convergence speed can be faster. After initialization, it generally only takes 10 iterations to converge, and the accuracy meets the needs of unmanned navigation.

c. The local positioning result of the lane line is invalid.

When lane line detection fails or false detection occurs, in order to maintain the stability of the system, the SLAM output result is used to calculate the current estimated value, so as to obtain the distance between the current position and the lane line in the map and generate the trajectory of the candidate points. At the same time, in the subsequent probability calculation link, this error is always 0, which does not affect the positioning result. The position update effect can still be achieved through the angle, position and other information. When the detection information is restored, the positioning function continues to be realized. Robustness is improved.

Finally, the algorithm obtains the current optimal positioning result. It is worth noting that this method can not only have a relocation effect on the global position, but also increase the positioning accuracy. At the same time, the results obtained for the local positioning of the lane lines can also have an optimization effect. Different from the results obtained directly from the local positioning of the lane lines detection, this paper calculates the local positioning by calculating the distance between the final output point with the maximum probability and the original lane line in the map. As a result, with the continuous information output by SLAM, the robust kernel function can ensure the normal operation of the system in the case of lane line detection failure or detection error. At the same time, the local positioning results are coherent and the accuracy is improved.

The algorithm proposed in this paper is essentially a mapping search for two trajectories, but different from the previous trajectory relocation algorithm, this paper applies the prediction model and successfully solves the real-time problem. It has great advantages when there is a small displacement or the position itself does not change significantly, especially when the convergence speed is greatly improved, which is suitable for real-time unmanned vehicle positioning.

It is worth noting that the algorithm proposed in this paper needs to adjust the parameters adaptively in different applications. The SLAM system used in this paper has a release frequency of 20 Hz, and the update frequency is equal to the release frequency of the SLAM system. There are points stored in the directed graph in the map. The interval is selected as 5 cm, and this value is the resolution of the positioning longitudinal error at different speeds and update frequencies. The value of K and the initial covariance matrix need to be adjusted. If the value of K is too large, the amount of calculation will increase. If the value of K is too small, it is easy to be limited to a specific area when updating, and the prediction incentive is not enough to break the local optimum. Similarly, the selection of the covariance matrix also affects the positioning results. In this paper, the value of K is selected as 10, and in the covariance matrix,  $(d_x, d_y)$  is set to 1,  $d_\theta$  is set to 0.2, and  $d_l$  is set to 2. With a selection interval of 5 cm, each update can affect about 20 map points, and the calculation speed is 5 ms. When the movement speed is significantly reduced or the publishing frequency is greatly increased, the K value needs to be adjusted. This paper provides an empirical conclusion. When the number of map points that can be affected by each update is about twice the value of K, the positioning effect is improved.

#### 4. Experimental Verification

The platform was modified from the Junipai A70E electric vehicle, and the body was loaded with sensors such as IMU, GNSS and a monocular camera. The IMU sensor adopts



MEMS products from Xsens, and the GNSS receiver adopts Huace P3 series products, and adopts the centimeter-level positioning service of Thousand Seek. Further, unmanned experiments were carried out on the Qingshuihe campus of the University of Electronic Science and Technology of China and the Longchi Tunnel. As shown in Table 1.

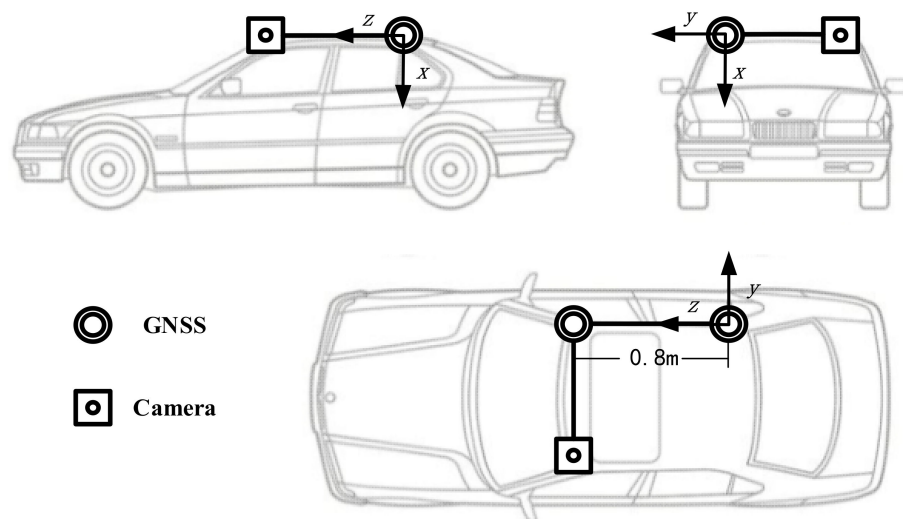
**Table 1.** Experimental platform hardware.

Hardware Name	Hardware Configuration	Features
GNSS positioning terminal	CTI P3, centimeter-level positioning accuracy	Cognitive map creation for CRRC’s satellite positioning as a visual sensing device (monocular)
Camera	LI-USB30-AR023ZWDR	
Vehicle main control unit	Industrial control workstation with graphics card, model 2080Ti	Supports multi-threaded running of code
In-vehicle peripherals	Monitor, etc.	Man-machine interface

In the experiment, the monocular camera + IMU sensor installed on the roof was used to complete the SLAM system. The body has sensors such as GNSS [24], IMU, and a camera. The GNSS dual-antenna device applied in the paper is used for true value positioning and orientation in unmanned driving experiments. Due to the dual antennas used, only the yaw angle and pitch angle of the vehicle can be calculated during orientation, but the roll angle cannot be calculated. Considering that the change of the roll angle of the vehicle body is small, it can be ignored in the actual application process. Most of the poses collected by GNSS represent the current position of the vehicle, so the GNSS antenna is required to be installed as parallel to the vehicle as possible. Figures 6 and 7 below shows the relative positional relationship between the sensors fixed on the roof.



**Figure 6.** (Top left–bottom right) Experimental equipment, sensor layout, campus scene, Longchi tunnel scene.



**Figure 7.** (Top left–bottom right) Experimental equipment, sensor layout, campus scene, Longchi tunnel scene.

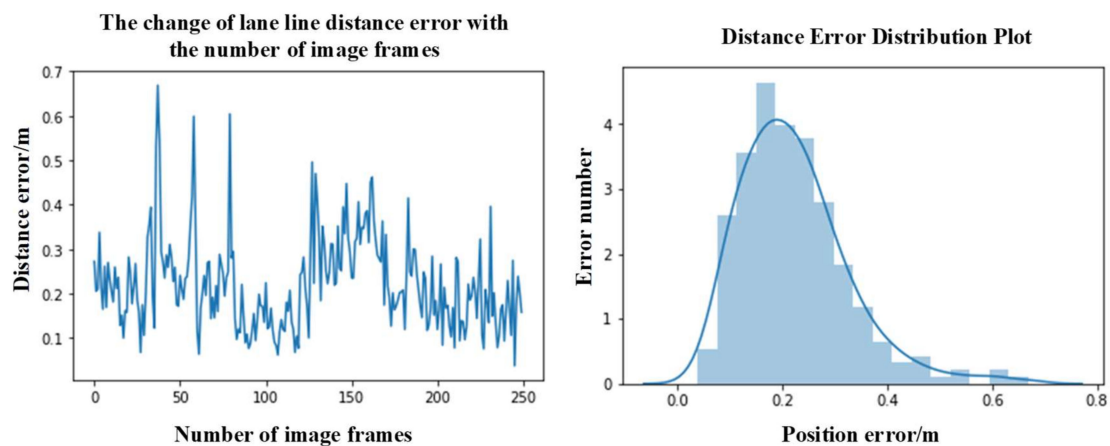
To evaluate the algorithm performance, an experimental analysis was performed. The work of map structuring can be stored in the map by the data collection vehicle and processed and obtained. It can also be completed by querying and interpolating the latitude and longitude on the Baidu map. Due to the basic deviation of the network map, the map data can be collected at the location with GNSS alignment.

The experimental data are that the data acquisition vehicle runs in the campus environment and the lane line area in the Longchi tunnel environment. This collects images and GNSS data. In order to judge the accuracy and performance of the algorithm, the true value data of the lane line position are needed. In this paper, the true value data of the lane line position were collected by a special trajectory acquisition vehicle. After the data were collected, the image data were used as the input of the algorithm, and the data collected by GNSS were used as the real value to evaluate the performance index of the algorithm. In the tunnel of the Longchi scene, there is a lack of real positioning value. This paper used the processed latitude and longitude interpolation results in the map as the real value, and fused the output positioning results and map data for trajectory error calculation.

#### 4.1. Experiment Results of Lane Line Positioning

In this paper, the monocular camera was used to calculate the relative position of the left and right lane lines, and the lane line positioning was determined. The true value was obtained by the data acquisition vehicle. The monocular camera positioning results of the lane line itself are as follows.

The error of distance  $D$  has no obvious change with the number of frames, and the overall error is relatively stable. According to a statistical analysis of all the error data, as shown in the right of Figure 8 above, the error distribution conforms to a skewed normal distribution, and most of the error distribution is within 0.3 m. The reliability of the vehicle position assumption used in this paper is generally stable.



**Figure 8.** Distance  $D$  positioning error. (left): the change of error with the number of frames; (right): the distribution of distance error.

#### 4.2. Campus Scene Localization Experiment Results

This experiment was carried out in the campus area. The real map lane value was obtained by the data acquisition car driving along the lane, and it can be considered that there is no map lane information error. GNSS was used as the true value of the data to calculate the original SLAM method, the SLAM algorithm with relocation information, and the positioning results of lane line positioning at this time, and to compare them with the true value of GNSS. The positioning results are as follows.

As shown in Figure 9, the two lines represent the true value of GNSS and the result obtained by the original SLAM algorithm. Here, the algorithm adopts the VINS-Mono algorithm, and the positioning result is displayed on the map after the coordinate system transformation. The true value of the map is displayed. It can clearly be seen from Figure 9

that the effect of the algorithm in this paper is better than the original SLAM method in practical application, and its positioning results basically coincide with the true value. It has a great influence, so the experiment is divided into two motion states of straight line and non-pure straight line.



Figure 9. On-campus location results.

When the lane is not a pure straight road, that is, the road has a radius, the  $Q$  matrix in the equation can be updated to obtain an accurate estimate of the position, and the vehicle trajectory and the road trajectory can be fused by adjusting the coefficient. The results of the positioning experiment and the schematic diagram are shown in Figure 10.

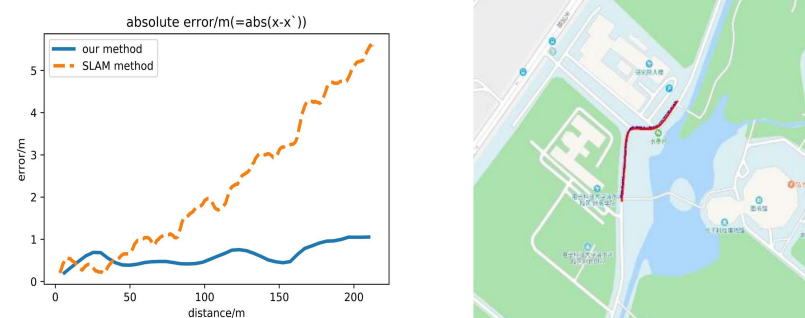


Figure 10. Detour test absolute positioning error.

In detours, the divergence of SLAM positioning results is better limited, the cumulative error is eliminated, and the positioning results of <1 m are obtained in gentle detours, and the effect is good.

As shown in Figure 11, In the straight-line experiment, since the lane line positioning information does not change greatly, the state quantity of the positioning cannot be corrected by observation, and the fusion positioning result at this time is expressed as the output lag of SLAM. The method proposed in this paper relies on the effective map information. When the pure linear motion has a lack of effective observation, the error cannot be corrected.

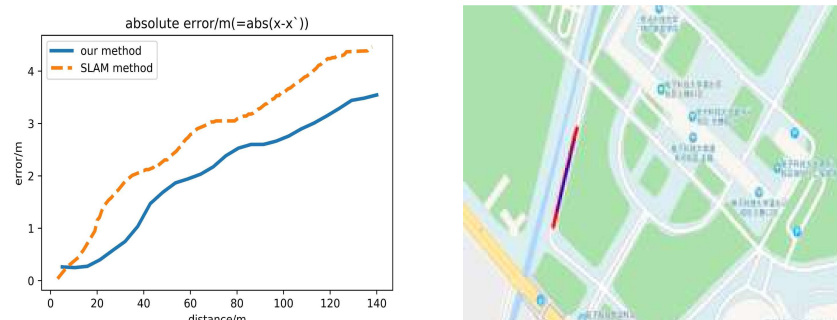
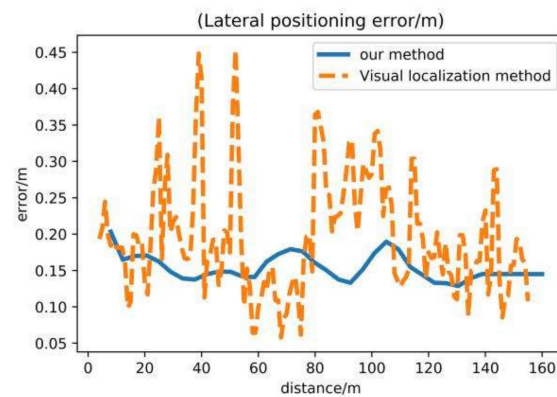


Figure 11. Absolute positioning error of straight road test.

However, in pure linear motion, although the error cannot be corrected by map information, the lateral positioning accuracy can still meet the needs of vehicle navigation. The results of lateral local positioning information are shown in Figure 12.



**Figure 12.** Lateral localization error.

As shown in Figure 12, the local positioning error does not exceed 0.3 m in most cases. When moving in a straight line, the lane can be cruised through local positioning. When encountering a detour, due to the fast-converging observation method set in the previous article, it can quickly converge to the location where it is located.

To summarize, the algorithm proposed in this paper has a good positioning effect on non-pure straight roads, and the positioning accuracy is greatly improved, while in pure straight line motion, the algorithm accuracy is less improved, but the results of lateral local positioning meet the requirements of unmanned driving. As shown in Table 2, the numerical analysis is as follows:

**Table 2.** Error comparison.

	Local Average Positioning Error/m	Local Maximum Positioning Error/m	Global Average Positioning Error/m
Original SLAM algorithm	1.3	2.1	5.94
Lane line localization algorithm	0.25	0.44	None
Combined with map SLAM relocation algorithm	0.16	0.36	1.35

As shown in the table above, in the data experiment of detours, the localization algorithm after integrating map information achieved good results, the global positioning accuracy improved by more than 70%, and the local positioning was also significantly improved, which is more suitable for behavior scenarios involving unmanned positioning use.

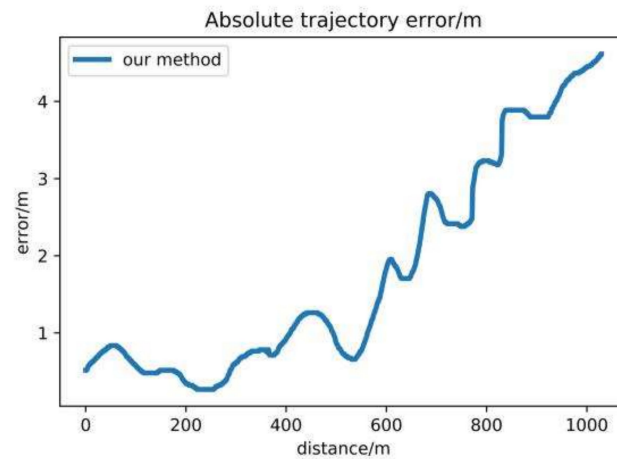
#### 4.3. Experiments Results in Closed Loop Path

As shown in Figure 13, The data set was collected at the south gate of the University of Electronic Science and Technology for the location experiment. The data were collected using the same experimental platform as in the previous section. The platform was equipped with GNSS terminals, cameras and other equipment to collect the dataset in data collection vehicle mode to obtain image data and GNSS position data during vehicle operation.

Compared to the original slam algorithm, our algorithm error is greatly improved, and the lane line positioning algorithm is able to constrain the lateral position of the vehicle relative to the lane line with good accuracy, eliminating most of the vehicle lateral position error. As shown in Table 3.

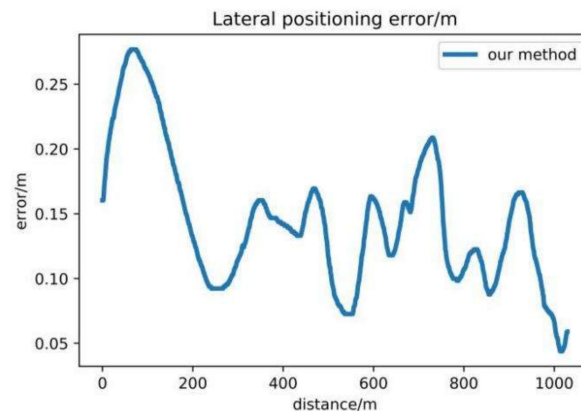


effect of the map is small, and the accumulated error is difficult to eliminate. After driving 600 m, the error reached m level, but the lateral local positioning results are still relatively stable. The distance between the driving trajectory and the road trajectory is calculated in the direction as the lateral positioning error.



**Figure 15.** Tunnel positioning trajectory error.

As shown in Figure 16, the lateral error positioning errors are all less than 0.35 m, and the initial positioning error is the error of the lane line positioning itself. Through the recalculation of the fusion positioning results, the positioning accuracy has been greatly improved and can be stabilized around 15 cm, which is sufficient. The unmanned vehicle completes the positioning and navigation in the fixed tunnel.



**Figure 16.** Tunnel positioning transverse tangent error.

It is worth noting that since there is no true value in the tunnel itself, the above two calculation methods are an approximate analysis, and the error between the true value and the positioning result is theoretically close to this method. Since the vehicle trajectory is smooth, the tunnel angle changes less, and the distance between the tangential directions and the actual point is less.

Through the above analysis, there are no GNSS scenarios such as tunnels. After the fusion of map lane information and the SLAM positioning method, the positioning accuracy has been greatly improved, and it has a good effect in global and local positioning, as well as better stability. The algorithm proposed in this paper can effectively complete the navigation under local positioning. Compared with the SLAM algorithm itself, the global positioning accuracy is greatly improved, which has great advantages for actual navigation.

This paper gives a summary of the following rules:

1. When the method in this paper moves in a pure straight line, the effect of the fusion positioning is the worst, which is basically only the lag of the SLAM output result.

2. The fusion of the method in this paper can effectively suppress the divergence of positioning errors and ensure the positioning requirements of unmanned vehicles. The lateral positioning results are always stable within 20 cm, which is sufficient for unmanned navigation in the local coordinate system.
3. Due to the continuity of the observation equation, the radius of curvature is small, but the position error with frequent changes is the smallest, which can be below the meter level.

## 5. Summary and Outlook

In this paper, by integrating SLAM positioning and map information, combined with the local stability and non-divergence of lane line positioning, and the globality and continuity of SLAM, a prior map of lane lines based on directed graphs is established to solve the problem of unmanned vehicle positioning. There is a lack of correlation between local positioning and global positioning, and the positioning results are not robust enough, especially in the case of GNSS failure, providing good local positioning information to ensure the realization of the navigation function. While preventing the divergence of global information, the positioning application of unmanned vehicles have a certain contribution. At the same time, the positioning method described in this paper can be applied to more scenes, such as semantic map information, the image information of known positioning, etc. By establishing observation equations and judging and adjusting one's own position, the accuracy of positioning can be increased further. The method proposed in this paper has practical application significance in universal map construction and localization. However, it is highly dependent on the observability of road information, and the fusion effect is limited when the road is a pure linear motion, which needs further research.

Driverless positioning technology is developing rapidly, and more sensors and algorithms are being proposed. Sensors such as millimeter wave radar, ultrasonic radar and infrared are all important in driverless positioning technology. SLAM systems are one of the important methods of multi-source sensor fusion, and we believe that the combined application of deep learning technology and SLAM will be an important method for future driverless capability.

**Author Contributions:** Conceptualization, S.L.; Formal analysis, Y.Z.; Methodology, Shuguang Li and C.S.; Project administration, H.C.; Software, C.S.; Writing—original draft, Z.L.; Writing—review & editing, X.L. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by the Key R&D Projects of the Science & Technology Department of Sichuan Province of China under Grant 2021YFG0070.

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

## References

1. Stączek, P.; Pizoń, J.; Danilczuk, W.; Gola, A. A Digital Twin Approach for the Improvement of an Autonomous Mobile Robots (AMR's) Operating Environment—A Case Study. *Sensors* **2021**, *21*, 7830. [[CrossRef](#)] [[PubMed](#)]
2. Brumerčik, F.; Lukac, M.; Caban, J. Unconventional powertrain simulation, Communications. *Sci. Lett. Univ. Žilina* **2016**, *18*, 30–33.
3. Neven, D.; de Brabandere, B.; Georgoulis, S.; Proesmans, M.; van Gool, L. Towards End-to-End Lane Detection: An Instance Segmentation Approach. In Proceedings of the 2018 IEEE Intelligent Vehicles Symposium (IV), Changshu, China, 26–30 June 2018; pp. 286–291.
4. van Gansbeke, W.; Neven, B.d.D.; Proesmans, M.; van Gool, L. End-to-end Lane Detection through Differentiable Least-Squares Fitting. In Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW), Seoul, Korea, 27–28 October 2019; pp. 905–913.
5. Redmon, J.; Farhadi, A. YOLOv3: An Incremental Improvement. *arXiv* **2018**, arXiv:1804.02767.
6. Sun, X.; Wu, P.; Hoi, S.C.H. Face Detection using Deep Learning: An Improved Faster RCNN Approach. *Neurocomputing* **2018**, *299*, 42–50. [[CrossRef](#)]
7. Zhou, L.-S. *Research on Lane Keeping Warning System Based on Monocular Vision*; East China University of Science and Technology: Shanghai, China, 2011.

8. Zhou, X.; Xiyue, H.; Yu, L. Monocular Vision-Based Highway Lane Keeping and Distance Measurement. *Chin. J. Graph.* **2003**, *110*–115.
9. Jianmin, D.; Yue, G.; Boyang, Z. A method for segmentation of drivable area based on binocular vision. *Electron. Meas. Technol.* **2019**, *42*, 138–143.
10. Davison, A.J.; Reid Ian, D.; Molton Nicholas, D.; Olivier, S. MonoSLAM: Real-time single camera SLAM. *IEEE Trans. Pattern Anal. Mach. Intell.* **2007**, *29*, 1052–1067. [[CrossRef](#)] [[PubMed](#)]
11. Klein, G.; Murra, D. Parallel Tracking and Mapping for Small AR Workspaces. In Proceedings of the IEEE ACM International Symposium on Mixed Augmented Reality, Nara, Japan, 13–16 November 2007.
12. Engel, J.; Schps, T.; Cremers, D. LSD-SLAM: Large-scale direct monocular SLAM. In Proceedings of the European Conference on Computer Vision, Zurich, Switzerland, 6–12 September 2014.
13. Forster, C.; Pizzoli, M.; Scaramuzz, D. SVO: Fast semi-direct monocular visual odometry. In Proceedings of the IEEE International Conference on Robotics Automation, Hong Kong, China, 31 May–7 June 2014.
14. Mur-Artal, R.; Tardós, J.D. ORB-SLAM2: An Open-Source SLAM System for Monocular, Stereo, and RGB-D Cameras. *EEE Trans. Robot.* **2017**, *33*, 1255–1262. [[CrossRef](#)]
15. Qin, T.; Li, P.; Shen, S. VINS-Mono: A Robust and Versatile Monocular Visual-Inertial State Estimator. *IEEE Trans. Robot.* **2018**, *34*, 1004–1020. [[CrossRef](#)]
16. Kasyanov, A.; Engelmann, F.; Stückler, J.; Leibe, B. Keyframe-based visual-inertial online SLAM with relocalization. In Proceedings of the 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Vancouver, BC, Canada, 24–28 September 2017; pp. 6662–6669.
17. Pengfei, D. *Research on the Key Technology of Monocular Vision SLAM Based on Graph Optimization*; Zhejiang University: Hangzhou, China, 2018.
18. Yiran, Z.; Chengjun, G. *A Deviation Correction Algorithm for Vehicle Navigation Trajectory Based on GPS and Map Matching*; China Satellite Navigation System Administration Office: Beijing, China, 2014; pp. 96–100.
19. Lianos, K.N.; Schönberger, J.L.; Pollefeys, M.; Sattler, T. VSO: Visual Semantic Odometry. In *Computer Vision—ECCV 2018*; Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y., Eds.; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2018; p. 11208.
20. Bowman, S.L.; Atanasov, N.; Daniilidis, K.; Pappas, G.J. Probabilistic data association for semantic SLAM. In Proceedings of the 2017 IEEE International Conference on Robotics and Automation (ICRA), Las Vegas, NV, USA, 29 May–3 June 2017; pp. 1722–1729.
21. Lees, A.C. Splits, lumps and shuffles. *Neotrop. Bird.* **2010**, *7*, 31–38.
22. Duyvené, D.W.T.; Eva, H.; Ulrich, J.; Gareth, E.; Carin, S. H-Net, the European Network for Harmonization of Training in Hematology, and its policy. *Haematologica* **2012**, *97*, 1776–1778.
23. Chum, O.; Matas, J.; Kittler, J. Locally Optimized RANSAC. *Pattern Recognition. Lect. Notes Comput. Sci.* **2003**, *2781*, 236–243.
24. Jagelčák, J.; Gnap, J.; Kuba, O.; Frnda, J.; Kostrzewski, M. Determination of Turning Radius and Lateral Acceleration of Vehicle by GNSS/INS Sensor. *Sensors* **2022**, *22*, 2298. [[CrossRef](#)] [[PubMed](#)]