

## Article

# Research on Positioning Method in Underground Complex Environments Based on Fusion of Binocular Vision and IMU

Jie Cheng <sup>1</sup>, Yinglian Jin <sup>2</sup>, Zhen Zhai <sup>1</sup>, Xiaolong Liu <sup>3</sup> and Kun Zhou <sup>1,\*</sup>

<sup>1</sup> School of Mechanical and Electrical Engineering, China Jiliang University, Hangzhou 310018, China; cj1426163582@163.com (J.C.); zhaizhen2022@163.com (Z.Z.)

<sup>2</sup> College of Modern Science and Technology, China Jiliang University, Hangzhou 310018, China; jinyinglian@cjlu.edu.cn

<sup>3</sup> Department of Mechanical Engineering, Johns Hopkins University, Baltimore, MD 21211, USA; xiaolong@jhu.edu

\* Correspondence: 13488950521@163.com or zk@cjlu.edu.cn

**Abstract:** Aiming at the failure of traditional visual slam localization caused by dynamic target interference and weak texture in underground complexes, an effective robot localization scheme was designed in this paper. Firstly, the Harris algorithm with stronger corner detection ability was used, which further improved the ORB (oriented FAST and rotated BRIEF) algorithm of traditional visual slam. Secondly, the non-uniform rational B-splines algorithm was used to transform the discrete data of inertial measurement unit (IMU) into second-order steerable continuous data, and the visual sensor data were fused with IMU data. Finally, the experimental results under the KITTI dataset, EUROCC dataset, and a simulated real scene proved that the method used in this paper has the characteristics of stronger robustness, better localization accuracy, small size of hardware equipment, and low power consumption.

**Keywords:** SLAM (simultaneous localization and mapping); sensor fusion of visual and IMU; nonlinear optimization; sliding window



**Citation:** Cheng, J.; Jin, Y.; Zhai, Z.; Liu, X.; Zhou, K. Research on Positioning Method in Underground Complex Environments Based on Fusion of Binocular Vision and IMU. *Sensors* **2022**, *22*, 5711. <https://doi.org/10.3390/s22155711>

Academic Editor: M. Osman Tokhi

Received: 8 July 2022

Accepted: 29 July 2022

Published: 30 July 2022

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



**Copyright:** © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

With the continuous expansion of urban scale and the continuous development of urban underground space, urban underground complex plays a more and more important role in urban development. At present, urban underground complexes mainly include rail transit stations, underground commercial streets, underground parking lots, and underground building entrances [1]. Underground commercial streets are places with high fire incidence, and the exhaust gas concentration of an underground parking lot is high. It is very important to design mobile robots to replace manual inspection in underground complex. The synchronous location and mapping (SLAM) technology is the key to realizing the autonomous navigation and inspection of mobile robots. Slam technology aims to use the self-loaded sensors to estimate their own position and posture, as well as to dynamically build a real-time map of the surrounding environment in an incremental way. After nearly 20 years of development, slam technology has played an important role in the fields of autonomous driving, unmanned driving, virtual reality, augmented reality, unmanned aerial vehicles, and so on. At present, the sensors used in the slam system mainly include camera, lidar, and inertial measurement units (IMU), and have two directions: laser slam and visual slam.

In 2014, Zhang and Singh proposed the LOAM (LIDAR odometry and mapping) algorithm [2], which obtained better pose estimation and 3D maps by calculating the curvature of each point in the point cloud and extracting the plane and edge features in the point cloud. In 2016, Google opened the laser SLAM algorithm Cartographer [3], which proposed the idea of local submap, which greatly accelerated the speed of loop-closure.

In 2018, Shan and others proposed the Lego-LOAM [4], which proposed the idea of clustering based on LOAM. In inter frame matching, only the points of the same cluster other than the ground cluster are matched, which increases the processing efficiency and ensures high accuracy. In the same year, Kim and others proposed SC-Lego-LOAM and added the Scan Context algorithm [5] on the basis of Lego-LOAM, which made up for the defect of poor loop-closure effect of Lego-LOAM. In 2020, Shan and others proposed the LIO-SAM algorithm [6]. LIO-SAM is a tightly coupled lidar inertial odometer framework based on factor graph, which is suitable for multi-sensor fusion and global optimization. It proposes an efficient scanning matching method based on the local sliding window, which improves the real-time performance. The above methods [2–6] belong to laser-based SLAM. LSD-SLAM (large-scale direct monocular SLAM) [7] proposed by Engel and others. It is a SLAM system based on direct method. The system is suitable for large-scale scenes and building large-scale and globally consistent environment maps. The visual odometer based on DSO (direct sparse odometry) [8] sparse direct method is superior to LSD-SLAM in accuracy, stability, and speed. The semi-direct visual odometer SVO (semi-direct visual odometry) was proposed by Forster et al. [9]. It combines the advantages of the method based on feature points and the direct tracking optical flow method, generating a lightweight environment map. The ORB-SLAM framework is a popular visual framework of the feature point method at present [10]. The front end is based on ORB features, which not only reduce the amount of calculation but also have higher accuracy than the direct method. On the basis of this framework, researchers have proposed several improved versions. ORB-SLAM2 is the first open-source SLAM algorithm supporting monocular, binocular, and RGB-D cameras; the BA method used at the back end of the algorithm can accurately estimate the scale of the trajectory [11]. Carlos and others proposed ORB-SLAM3 in 2021 [12]. The multi-map system of the algorithm enabled the algorithm to maintain operation when visual information was lost for a long period of time. The above methods belong to visual-SLAM. However, there are a large number of mobile people in the underground complex, which seriously interferes with the information collected by laser sensors and visual sensors. The texture of the building facing materials of the underground complex is simple, and the environmental features are sparse, so it is difficult to extract the features. As a result, the traditional laser slam and visual slam are difficult to be directly applied to the scene of an underground complex.

Aiming at the dynamic target interference and weak texture characteristics of underground complex, a robot positioning method based on binocular vision and inertial navigation (IMU) fusion is designed in this paper.

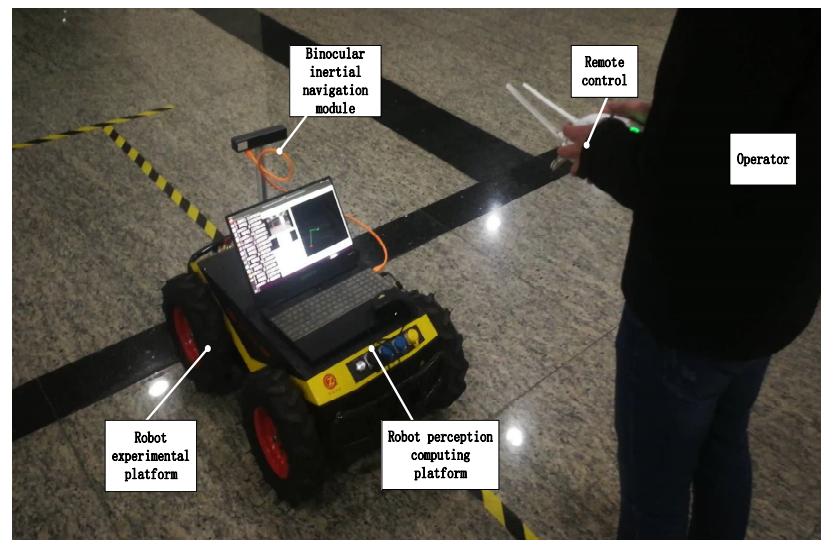
The main contributions of this paper are provided as follows:

- (1) The combination of the Harris algorithm and the optical flow pyramid tracking method enhances the robustness of the visual positioning system and then solves the problem of the front end being difficult in terms of extracting feature points in weak texture and it being easy to lose tracking.
- (2) The interpolation algorithm of non-uniform rational B-spline is used to transform the discrete data of IMU into a continuous track that can be second-order differentiated, to align the visual and inertial information, and to improve the anti-dynamic interference ability by integrating the visual and inertial data.

The remaining parts of this paper are organized as follows: Section 2 mainly introduces the hardware framework. Section 3 presents the core algorithm design of binocular vision and interval navigation fusion. Some practical experiments are presented in Section 4 to demonstrate the effectiveness of the obtained results. The last section concludes this paper.

## 2. Hardware Framework

As shown in Figure 1, the hardware framework used in this paper is mainly composed of the binocular inertial navigation module, the inspection robot (chassis), the remote control, and the robot perception computing platform.



**Figure 1.** Hardware framework of robot.

The camera uses the small binocular inertial navigation integrated module and uses a 120 degree wide angle lens to obtain environmental information as much as possible. The IMU is a six-axis BMI088 sensor manufactured by BOSCH with a synchronization accuracy of 0.05 ms.

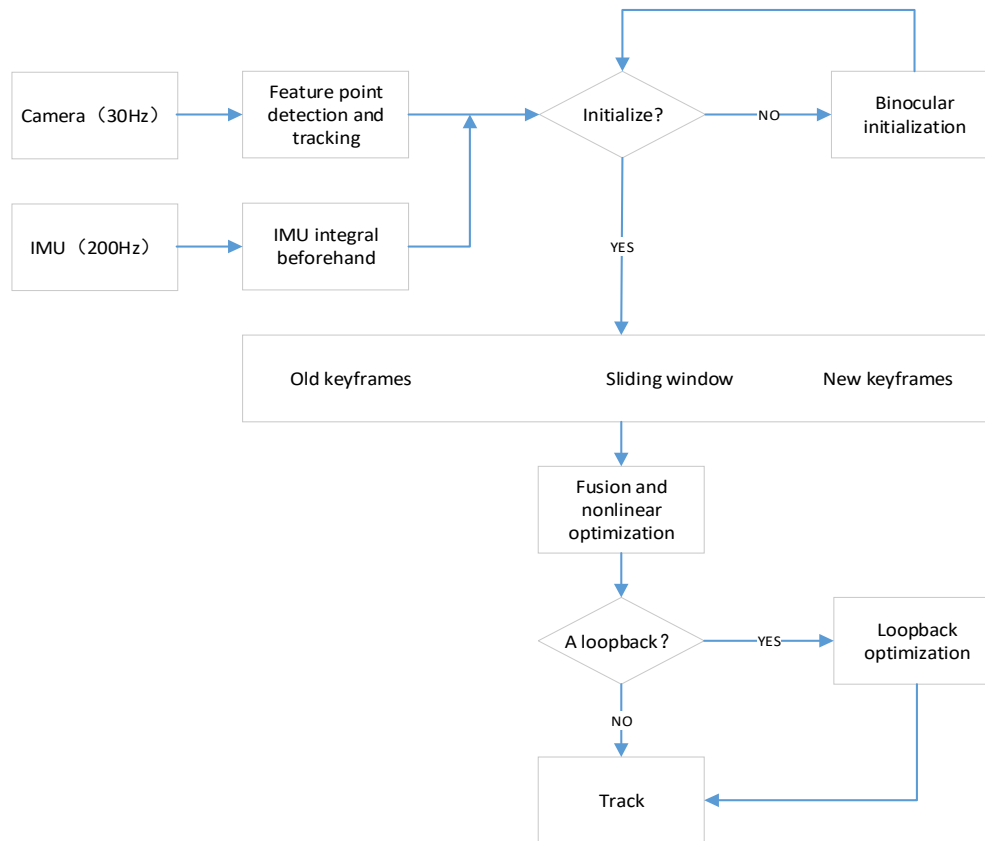
The computing platform is mainly responsible for processing sensor data, using Intel(R) Core(TM) I7-7700HQ CPU, 4 cores, 8 threads, 16 GB LPDDR4 memory; the main frequency is 2.80 ghz. The operating system is Ubuntu18.04 LTS, in which ROS (robot operating system) is installed to deal with the communication of various sensor data and the operation of specific algorithms.

### 3. Core Algorithm Design of Binocular Vision and Interval Navigation Fusion

In this paper, the positioning method of visual fusion inertial navigation designed is mainly composed of the following three parts:

- (1) Through the study of traditional visual odometers, it is found that the front end based on ORB feature points is not suitable for a weakly textured environment. The improved Harris algorithm with a stronger ability to extract corner points ensures that the front end can extract more feature points from weak texture.
- (2) Due to the high sampling frequency of the inertial navigation system, it is difficult to align and fuse high-frequency data with low-frequency data. To solve these problems, firstly, we studied and analyzed various factors of inertial navigation error, simplified the inertial navigation model, and reduced the difficulty of calculation. Secondly, non-uniform rational B-spline algorithm was used to transform high-frequency discrete IMU data into a second-derivable continuous trajectory, which is convenient for fusion with visual information.
- (3) As for integration of vision and inertial navigation, first, since every update of back-end optimization needs to recalculate the inertial derivative data in the world coordinate system, the original inertial derivative data can be preprocessed to obtain the pre-integral, which can reduce the number of repeated calculation and deduce the Jacobian matrix and covariance. Secondly, visual and inertial navigation information were integrated in a tightly coupled way to construct optimization objective function and state variables. Thirdly, a more reasonable Levenberg–Marquardt method was introduced to solve the non-singular ill-condition problems in the fusion optimization problem. Fourthly, sliding windows and marginalization were adopted to delete the old state quantity and release the computing power in view of the growing computing scale of the back end.

The above steps can be represented by the following flow chart (Figure 2):



**Figure 2.** The algorithmic framework used in this paper.

### 3.1. Binocular Vision Algorithm

In this paper, the binocular vision algorithm mainly consisted of front end, back end, and loop-closure.

In the front end, the following method was used to obtain the camera pose transformation between two adjacent frames:

- (1) Extract Harris feature points of the first frame [13].
- (2) LK multilayer optical flow was used to calculate the corresponding feature points in the second frame [14].
- (3) DLT (direct linear transformation) was used to calculate the pose transformation of the camera through matched pairs of feature points [15].

In the back end, BA (bundle adjustment method) was adopted to modify the existing camera pose and roadmap [16].

In the aspect of loop-closure, the bag-of-words (BoW) model was adopted, and the steps are as follows:

- (1) A number of pictures around the field scene were collected in advance, FAST feature points of each image were extracted, BRIEF descriptors of feature points were calculated to construct words of the image [17], DBoW3 was used as a dictionary to store words [18,19], and the TF-IDF algorithm was used to add different weights to different words [20].
- (2) At the same time, for each image obtained, the characteristics of the image were described as words in BoW, the frequency of words was counted, and an image was converted into a group of words.

The similarity of words in each image is calculated to determine whether loop-closure is triggered.

### 3.2. Inertial Navigation Algorithm

The inertial navigation system has a stronger anti-interference ability compared with the visual positioning system, but there are many factors affecting the error that will lead to the drift of the inertial measurement value and the accumulated error being too large. This paper introduces the basic content of inertial navigation and lays the foundation for the subsequent multi-sensor fusion.

#### 3.2.1. Inertial Error Model

The errors of the accelerometer and gyroscope in inertial navigation (IMU) can be divided into deterministic errors and random errors. Deterministic errors can be obtained through prior calibration, which mainly includes bias errors, scale errors, and axis misalignments [21]. For random error, it is assumed that the noise obeys Gaussian distribution, including Gaussian white noise and bias random walk. The formula of the total error model can be expressed as

$$\tilde{\omega}^b = S_g \omega^b + b^g + n^g \quad (1)$$

$$\tilde{a}^b = S_a q_{bw} (a^w - g^w) + b^a + n^a \quad (2)$$

In Formula (1),  $\tilde{\omega}^b$  and  $\omega^b$  represent the measured value and the real value of angular velocity in the body coordinate system of IMU, respectively;  $b^g$  and  $n^g$  denote the bias random walk and Gaussian white noise of the gyroscope, respectively; and  $S_g$  is the scale factor of the gyroscope. In Formula (2),  $\tilde{a}^b$  is the measured value of acceleration in the body coordinate system of IMU;  $a^w$  and  $g^w$  are the world coordinates of real value and gravity acceleration, respectively;  $b^a$  and  $n^a$  denote the bias random walk and Gaussian white noise of the accelerometer, respectively;  $S_a$  is the scale factor of the accelerometer; and  $q_{bw}$  denotes the rotation transformation from body to world coordinates.

#### 3.2.2. Inertial Derivatives Calculated

There are usually two methods to solve IMU data, namely, the Euler method and the median method.

The Euler method assumes that the derivative of the integrand function is constant in a short period of time. When calculating the pose at  $k+1$ , the linear velocity and angular velocity used are those measured at  $k$ , and the formula can be designed as

$$P_{wb_{k+1}} = P_{wb_k} + v_k^w \Delta t + \frac{1}{2} a \Delta t^2 \quad (3)$$

$$v_{k+1}^w = v_k^w + a \Delta t \quad (4)$$

$$q_{wb_{k+1}} = q_{wb_k} \otimes \left[ \begin{array}{c} 1 \\ \frac{1}{2} \omega \delta t \end{array} \right] \quad (5)$$

$$a = q_{wb_k} (a^{b_k} - b_k^a) - g^w \quad (6)$$

$$\omega = \omega^{b_k} - b_k^g \quad (7)$$

In the above formula,  $P_{wb_k}$  and  $P_{wb_{k+1}}$  are the position coordinates of the robot in the world coordinate system at time  $k$  and time  $k+1$ , respectively;  $v_k^w$  and  $v_{k+1}^w$  are the robot speed in the world coordinate system at time  $k$  and time  $k+1$ , respectively;  $a$  is the acceleration;  $\Delta t$  is the time interval between  $k$  and  $k+1$ ;  $q_{wb_k}$  and  $q_{wb_{k+1}}$  denote robot rotation information in the world coordinate system at time  $k$  and time  $k+1$ , respectively;  $\otimes$  stands for quaternion multiplication;  $a^{b_k}$  is the acceleration in the body coordinate system of IMU at time  $k$ ;  $b_k^a$  is the bias noise;  $g^w$  denotes acceleration of gravity;  $\omega^{b_k}$  is the angular velocity in the body coordinate system of IMU at time  $k$ ; and  $b_k^g$  denotes the bias noise.

Although the median method also assumes that the derivative of the integrand is constant in a short period of time, the linear velocity and angular velocity used in the calculation are the average values of the linear velocity and angular velocity measured at the moment  $k$  and  $k+1$ , and the formula can be expressed as

$$P_{wb_{k+1}} = P_{wb_k} + v_k^w \Delta t + \frac{1}{2} a \Delta t^2 \quad (8)$$

$$v_{k+1}^w = v_k^w + a \Delta t \quad (9)$$

$$q_{wb_{k+1}} = q_{wb_k} \otimes \begin{bmatrix} 1 \\ \frac{1}{2} \omega \delta t \end{bmatrix} \quad (10)$$

$$a = \frac{1}{2} \left[ q_{wb_k} (a^{b_k} - b_k^a) - g^w + q_{wb_{k+1}} (a^{b_{k+1}} - b_k^a) - g^w \right] \quad (11)$$

$$\omega = \frac{1}{2} \left[ (\omega^{b_k} - b_k^g) + (\omega^{b_{k+1}} - b_k^g) \right] \quad (12)$$

The new variables in the above formula are as follows:  $a^{b_{k+1}}$  is the acceleration in the body coordinate system of IMU at  $k+1$ , and  $\omega^{b_{k+1}}$  is the angular velocity in the body coordinate system of IMU at  $k+1$ .

Inertial navigation has a very high sampling frequency (usually between 100 and 200 Hz) and the raw data obtained from inertial navigation is huge. The computing platform mounted on the inspection robot has limited computing power, but it is expected that the calculation error is small. Therefore, a simulation experiment was designed to compare the performance of the two algorithms.

Suppose a particle is moving elliptically in the  $xy$  plane and sinusoidal in the  $z$  axis. The other particle performs the same motion as the first point, but adds the IMU kinematic model and outputs the values measured by the IMU in the simulation. According to the motion model of IMU in continuous time, the Euler method and median method are adopted. By comparing the error mean and time consumption of the Euler method and median method, an appropriate integration algorithm is selected.

Let the motion parameter equation of the particle be translated as

$$(15 \cos(\pi t/10 + 5), 20 \sin(\pi t/10), \sin(\pi t) + 5) \quad (13)$$

The rotation formula can be expressed as

$$(0.1 \cos(t), 0.2 \sin(t), \pi t) \quad (14)$$

where  $t$  is the time.

As shown in Figure 3, the blue curve is the true value of particle trajectory, while the yellow curve is the trajectory of a particle calculated by the Euler method or median method under the measurement of the IMU model. The mean error of the Euler method is 0.03099, and the mean error of median method is 0.02130. The latter is 31.27% higher than the former. The average calculation time of the two algorithms on 10 sets of data are 0.05571 s and 0.07061 s, respectively. The speed of the former is 21.10% faster than the latter. In the fusion algorithm, even if the initial pose is not accurate, the subsequent pose will continue to be optimized. However, time wasted cannot be recovered, which is an irreversible indicator. In view of the large number of computing modules in the system, it is necessary to allocate computing resources reasonably to ensure the real-time performance of the fusion system. Therefore, for the whole visual and inertial navigation integration system, this paper finally chose the Euler integral.

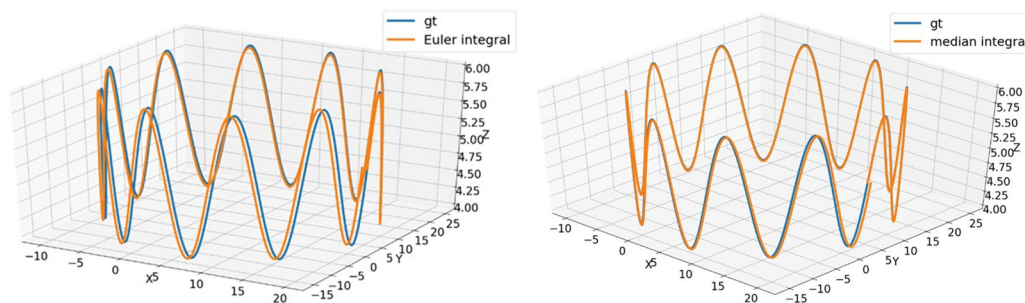


Figure 3. The comparison of Euler’s method (left) and median method (right).

### 3.2.3. Nonuniform Rational B Spline

The discrete position, velocity, and attitude information can be obtained by IMU data solution, and the position and attitude information obtained by visual odometer is also discrete. The problem of timestamp unsynchronization exists between two kinds of discrete data [22]. If a smooth, multi-order differentiable curve is generated according to the discrete visual odometer data, the angular velocity can be obtained by taking the first derivative of the curve, which can be compared with the angular velocity obtained by IMU solution, so as to realize the alignment between the visual odometer and IMU data.

The Bessel curve is usually used to series known discrete points to generate a continuous trajectory, but when there are too many points, adjusting any one point will affect the whole Bessel curve, and each point will have weaker control over the curve. On this basis, this paper improved and adopted the NURBS (non-uniform rational B-splines) method [23] to solve the above problems through multiple recursion of piecewise functions. Non-uniform means that there is a difference between each two node vectors, and rational means that different weights can be applied to each point; the formula can be expressed as

$$P(t) = \frac{\sum_{i=0}^n \omega_i P_i N_{i,k}(t)}{\sum_{i=0}^n \omega_i N_{i,k}(t)} \tag{15}$$

where  $t$  denotes the node,  $T = [t_0, t_1, \dots, t_{n+k}]$  means the set range,  $\omega_i$  means the weight factor,  $k$  is the degree of this curve,  $N_{i,k}(t)$  denotes the  $i$ -th and  $k$ -th B-spline basis function, and  $p_i$  means the  $i$ -th control point in the Bezier curve; the recursive formula (Cox-de Boor) is defined as

$$N_{i,0}(t) = \begin{cases} 1, & t_i \leq t < t_{i+1} \\ 0, & t \geq t_{i+1} \text{ or } t < t_i \end{cases} \tag{16}$$

$$N_{i,k}(t) = \frac{t - t_i}{t_{i+k} - t_i} N_{i,k-1}(t) + \frac{t_{i+k+1} - t}{t_{i+k+1} - t_{i+1}} N_{i+1,k-1}(t), k \geq 1 \tag{17}$$

Through the recursive logic of B-spline basis function, it can be found that each  $k$ -degree basis function only has values in  $n$  node intervals, which means that this control point achieves local control that the Bessel curve cannot achieve.

### 3.3. A Tightly Coupled Visual and Inertial Fusion Algorithm

The first two sections mainly introduce the method of pure visual positioning and pure inertial positioning and make some improvements for specific scenes. The main content of this section is to solve the problem of fusion of two types of non-homologous sensor data.

#### 3.3.1. IMU Data Preprocessing

IMU has a high sampling frequency and a large amount of data. In optimization problems, it is impossible to put so much data into state variables. Therefore, the usual practice is as follows: Given the translation, velocity, and attitude of the  $k$ -th s, and given all the data between the  $k$ -th sand the  $k+1$  s as well as the known dynamics equation,

integrating from the  $k$ -th s to the  $k+1$  s, we can finally obtain the translation, velocity, and attitude of the next moment. However, in the subsequent optimization algorithm, if the iterative solution calculation updates the state quantity in this way, every time the starting point of the iteration changes, or if any state quantity is adjusted midway, it must be re-integrated in order to obtain the following trajectory. Limited computing resources cannot support such calculations. To solve this problem, the IMU pre-integration algorithm is used [24–26], and the formula can be defined as

$$R_w^{b_k} P_{b_{k+1}}^w = R_w^{b_k} \left( p_{b_w}^w + v_{b_k}^w \Delta t_k - \frac{1}{2} g^w \Delta t_k^2 \right) + \alpha_{b_{k+1}}^{b_k} \tag{18}$$

$$R_w^{b_k} v_{b_{k+1}}^w = R_w^{b_k} \left( v_{b_k}^w - g^w \Delta t_k \right) + \beta_{b_{k+1}}^{b_k} \tag{19}$$

$$q_w^{b_k} \otimes q_w^{b_{k+1}} = \gamma_{b_{k+1}}^{b_k} \tag{20}$$

where  $R_w^{b_k}$  is the rotation transformation from IMU coordinate system to the world coordinate system at time  $k$ ;  $q_w^{b_k}$  is the quaternion form;  $P_{b_k}^w$  and  $P_{b_{k+1}}^w$  represent the position coordinates of IMU at  $k$  moment and  $k+1$  moment in the world coordinate system, respectively;  $v_{b_k}^w$  and  $v_{b_{k+1}}^w$  represent the velocity of IMU at time  $k$  and time  $k+1$  in the world coordinate system, respectively;  $g^w$  is the acceleration of gravity in the world coordinate system; and  $\Delta t_k$  is the interval between  $k$  moment and  $k+1$  moment; the definition of  $\alpha_{b_{k+1}}^{b_k}$ ,  $\beta_{b_{k+1}}^{b_k}$ , and  $\gamma_{b_{k+1}}^{b_k}$  are defined as follows

$$\alpha_{b_{k+1}}^{b_k} = \iint_{t \in [k, k+1]} \left[ R_t^{b_k} \left( \tilde{a}_t - b_{a_t} \right) \right] dt^2 \tag{21}$$

$$\beta_{b_{k+1}}^{b_k} = \int_{t \in [k, k+1]} \left[ R_t^{b_k} \left( \tilde{a}_t - b_{a_t} \right) \right] dt \tag{22}$$

$$\gamma_{b_{k+1}}^{b_k} = \int_{t \in [k, k+1]} \frac{1}{2} \Omega \left( \tilde{\omega}_t - b_{w_t} \right) \gamma_t^{b_k} dt \tag{23}$$

$$\Omega(\omega) = \begin{bmatrix} -[\omega]_{\times} & \omega \\ -\omega^T & 0 \end{bmatrix} \tag{24}$$

where  $\tilde{a}_t$  and  $\tilde{\omega}_t$  are acceleration and angular velocity given by IMU at time  $T$ , respectively;  $b_{a_t}$  and  $b_{w_t}$  are bias noises; and  $\alpha_{b_{k+1}}^{b_k}$ ,  $\beta_{b_{k+1}}^{b_k}$ , and  $\gamma_{b_{k+1}}^{b_k}$  are the pre-product components of translation, velocity, and attitude from frame  $b_k$  to frame  $b_{k+1}$  respectively.

IMU pre-integration can effectively reduce the amount of calculations. However, the noise variance of the original IMU data as the measured value can be obtained through calibration, but the uncertainty of the pre-integration is unknown now, so the covariance of the pre-integration needs to be derived.

In discrete form, the linear transfer of the incremental error at adjacent moments satisfies the equation and can be expressed as

$$\delta z_{t+\delta t}^{b_k} = F_t \delta z_t^{b_k} + G_t n_t \tag{25}$$

The parameters are as follows

$$z_t^{b_k} = [\delta \alpha_k \quad \delta \theta_k \quad \delta \beta_k \quad \delta b_{a_k} \quad \delta b_{w_k}]^T \tag{26}$$

$$n_t = [n_{a_k} \quad n_{w_k} \quad n_{a_{k+1}} \quad n_{w_{k+1}} \quad n_{b_a} \quad n_{b_w}]^T \tag{27}$$



$$F_t = \begin{bmatrix} I & f_{01} & I\delta t & f_{03} & f_{04} \\ 0 & f_{11} & 0 & 0 & -I\delta t \\ 0 & f_{21} & I & f_{23} & f_{24} \\ 0 & 0 & 0 & I & 0 \\ 0 & 0 & 0 & 0 & I \end{bmatrix} \quad (28)$$

$$G_t = \begin{bmatrix} v_{00} & v_{01} & v_{02} & v_{03} & 0 & 0 \\ 0 & -\frac{I\delta t}{2} & 0 & -\frac{I\delta t}{2} & 0 & 0 \\ -\frac{R_k\delta t}{2} & v_{21} & -\frac{R_{k+1}\delta t}{2} & v_{23} & 0 & 0 \\ 0 & 0 & 0 & 0 & I\delta t & 0 \\ 0 & 0 & 0 & 0 & 0 & I\delta t \end{bmatrix} \quad (29)$$

where  $z_t^{b_k}$  is a vector composed of five state increments, namely, pre-integrated translation, rotation, velocity, accelerometer bias noise, and gyroscope bias noise [27]. The details of parameters in matrix  $F_t$ ,  $G_t$  are shown in Table 1.

**Table 1.** The parameters in matrix  $F_t$ ,  $G_t$ .

Coefficient	Mathematical Formula
$f_{01}$	$-\frac{1}{4} \left( R_{b_i b_k} [a^{b_k} - b_k^a]_{\times} \delta t^2 + R_{b_i b_{k+1}} [(a^{b_k} - b_k^a)]_{\times} (I - [\omega]_{\times} \delta t) \delta t^2 \right)$
$f_{03}$	$-\frac{1}{4} (q_{b_i b_k} + q_{b_i b_{k+1}}) \delta t^2$
$f_{04}$	$-\frac{1}{4} \left( R_{b_i b_{k+1}} [(a^{b_k} - b_k^a)]_{\times} \delta t^2 \right) (-\delta t)$
$f_{11}$	$I - [\omega]_{\times}$
$f_{21}$	$-\frac{1}{2} \left( R_{b_i b_k} [a^{b_k} - b_k^a]_{\times} \delta t + R_{b_i b_{k+1}} [(a^{b_k} - b_k^a)]_{\times} (I - [\omega]_{\times} \delta t) \delta t \right)$
$f_{23}$	$-\frac{1}{2} (q_{b_i b_k} + q_{b_i b_{k+1}}) \delta t$
$f_{24}$	$-\frac{1}{2} \left( R_{b_i b_{k+1}} [(a^{b_k} - b_k^a)]_{\times} \delta t \right) (-\delta t)$
$v_{00}$	$\frac{1}{4} q_{b_i b_k} \delta t^2$
$v_{01}$	$-\frac{1}{4} \left( R_{b_i b_{k+1}} [(a^{b_k} - b_k^a)]_{\times} \delta t^2 \right) \left( \frac{1}{2} \delta t \right)$
$v_{02}$	$\frac{1}{4} q_{b_i b_{k+1}} \delta t^2$
$v_{03}$	$-\frac{1}{4} \left( R_{b_i b_{k+1}} [(a^{b_k} - b_k^a)]_{\times} \delta t^2 \right) \left( \frac{1}{2} \delta t \right)$
$v_{21}$	$-\frac{1}{2} \left( R_{b_i b_{k+1}} [(a^{b_k} - b_k^a)]_{\times} \delta t^2 \right) \left( \frac{1}{2} \delta t \right)$
$v_{23}$	$-\frac{1}{2} \left( R_{b_i b_{k+1}} [(a^{b_k} - b_k^a)]_{\times} \delta t^2 \right) \left( \frac{1}{2} \delta t \right)$

It can be seen from Formula (25) that error transmission can be divided into two types: error transmission from the current moment to the next moment and measurement noise transmission from the current moment to the next moment.

Here,  $\delta z_{t+\delta t}^{b_k}$  can be approximated by first-order Taylor, and the formula can be expressed as

$$\begin{aligned} \delta z_{t+\delta t}^{b_k} &= \delta z_t^{b_k} + \delta \dot{z}_t^{b_k} \delta t \\ &= (I + F_t \delta t) \delta z_t^{b_k} + (G_t \delta t) n_t \\ &= F \delta z_t^{b_k} + V n_t \end{aligned} \quad (30)$$

According to Formula (30), IMU measurement error is transitive, and the error at the current moment has a linear relationship with the covariance at the next moment, so the latter can be calculated from the former.

The calculation formula of covariance matrix can be expressed as

$$P_{t+\delta t}^{b_k} = (I + F_t \delta t) P_t^{b_k} (I + F_t \delta t)^T + (G_t \delta t) Q (G_t \delta t)^T \quad (31)$$

where the initial value of variance is  $P_0^{b_k} = 0$ , and  $Q$  is the diagonal covariance matrix of the noise term.

In addition, according to the formula, the iterative formula of the Jacobian matrix of the error term can be obtained

$$J_{t+\delta t} = (I + F_t \delta t) J_t \tag{32}$$

where  $J_t$  is the Jacobian matrix, and the initial value of the Jacobian matrix is the identity matrix.

The pre-integral values of IMU, covariance matrix, and Jacobian matrix of residuals are deduced above.

### 3.3.2. Construction of Residual Equation

In order to describe the difference between the two sensors, a nonlinear objective function needs to be constructed. The objective of optimization is to make the difference as small as possible. Optimization variables include camera information and IMU information. The initial value of iteration is the imprecise initial value measured by the sensor, and then optimization is carried out on this basis, so as to ensure that the optimized variable is the global optimal as much as possible and avoid falling into the local optimal solution.

(a) The state vector

The elements of the state vector include  $n + 1$  sensor state quantity, one external parameter, and inverse depth of  $m + 1$  space path punctuation [28]. The formula can be expressed as

$$X = [x_0, x_1, \dots, x_n, x_c^b, \lambda_0, \lambda_1, \dots, \lambda_m] \tag{33}$$

$$x_k = [p_{b_k}^w, v_{b_k}^w, q_{b_k}^w, b_a, b_g] \tag{34}$$

$$x_c^b = [p_c^b, q_c^b] \tag{35}$$

where  $X$  is the state vector, and  $x_k$  denotes the state quantity of the sensor in the  $k$ -th frame in the sliding window, in which there are  $n$  key frames in total. Each state quantity has five variables, namely, the translation of the  $k$ -th frame, the velocity quantity, the rotation quantity, and the deviation between the accelerometer and the gyroscope.  $x_c^b$  is the external parameter from camera to IMU. The inverse depth is the inverse of the signpost depth.

(b) Inertial constraint

Define the difference between IMU position, rotation, velocity, acceleration bias, and gyro bias changes every two frames, and the formula can be expressed as

$$r_B \left( \begin{matrix} \sim b_k \\ z_{b_{k+1}} \end{matrix}, X \right) = \begin{bmatrix} \delta \alpha_{b_{k+1}}^{b_k} \\ \delta \theta_{b_{k+1}}^{b_k} \\ \delta \beta_{b_{k+1}}^{b_k} \\ \delta b_a \\ \delta b_g \end{bmatrix} = \begin{bmatrix} R_{b_{k+1}}^{b_k} \left( p_{b_{k+1}}^w - p_{b_k}^w - v_{b_k}^w \Delta t_k + \frac{1}{2} g^w \Delta t_k^2 \right) - a_{b_{k+1}}^{b_k} \\ 2 \left[ \gamma_{b_{k+1}}^{b_k} \otimes q_{b_k}^{w-1} \otimes q_{b_{k+1}}^w \right]_{xyz} \\ R_{b_{k+1}}^{b_k} \left( v_{b_{k+1}}^w - v_{b_k}^w + g^w \Delta t_k \right) - \beta_{b_{k+1}}^{b_k} \\ b_{a_{b_{k+1}}} - b_{a_{b_k}} \\ b_{\omega_{b_{k+1}}} - b_{\omega_{b_k}} \end{bmatrix} \tag{36}$$

The optimization variables in the formula are the state quantity of frame  $k$  and the state quantity of frame  $k + 1$ , and there are 10 variables in total.

(c) Visual constraints

Visual residuals occur when pixels are reprojected. In order to obtain the ideal sensor pose, visual residuals need to be minimized. The coordinate value of the landmark point  $l$  observed in the camera coordinate system of frame  $i$  projected to the pixel is converted to the camera coordinate system of frame  $j$ , and the visual residual is composed of the pixel

coordinate value obtained by conversion and the directly observed pixel coordinate value, and the calculation formula can be expressed as

$$r_c(\tilde{z}_l^c, X) = \begin{bmatrix} \frac{x_{c_j}}{z_{c_j}} - u_{c_j} \\ \frac{y_{c_j}}{z_{c_j}} - v_{c_j} \end{bmatrix} \quad (37)$$

where  $x_{c_j}, y_{c_j}, z_{c_j}$  are the space coordinate value obtained by conversion, and  $u_{c_j}, v_{c_j}$  is the pixel coordinate value obtained by direct observation. The value of the feature point in frame  $i$  projected to the camera coordinate system of frame  $j$  can be defined as

$$\begin{bmatrix} x_{C_j} \\ y_{C_j} \\ z_{C_j} \\ 1 \end{bmatrix} = T_{bc}^{-1} T_{wb_j}^{-1} T_{wb_i} T_{bc} \begin{bmatrix} \frac{1}{\lambda} u_{C_i} \\ \frac{1}{\lambda} v_{C_i} \\ \frac{1}{\lambda} \\ 1 \end{bmatrix} \quad (38)$$

where  $\lambda$  is the inverse depth. The optimization variables are two moment state variables, external parameters and inverse depth.

#### (d) Global objective function

Building the overall objective function includes prior information, visual information, and IMU constraint information, and in integrating the information of various sensors into the nonlinear optimization to obtain the optimal state vector, the formula can be expressed as

$$\min_X \left\{ \|r_p - J_p X\|^2 + \sum_{k \in B} \left\| r_B(\tilde{z}_{b_{k+1}}^{b_k}, X) \right\|_{P_{b_{k+1}}^{b_k}}^2 + \sum_{(l,j) \in C} \left\| r_c(\tilde{z}_l^c, X) \right\|_{P_l^{C_j}}^2 \right\} \quad (39)$$

where  $P_l^{C_j}$  is noise covariance generated by visual observation, and the noise covariance generated by IMU pre-integration is  $P_{b_{k+1}}^{b_k}$ . Covariance is a weight that regulates the confidence of IMU and visual sensors in fusion. The more noise a sensor makes, the less reliable it is. Reduce the weight of this sensor and increase the weight of another sensor.

#### 3.3.3. Nonlinear Optimization Solution

After constructing the nonlinear objective function of vision and IMU fusion, solving this equation requires knowing the global properties of the objective function, but it cannot be solved directly. In this paper, both the Levenberg–Marquardt (L-M) method and the Gauss–Newton (G-N) method were used as the gradient descent strategy. When the problem is good, the Gauss–Newton method is used. If the problem approaches pathology, the Levenberg–Marquardt method is used [29].

In the actual fusion process of vision and IMU, if disturbed, the value of a certain residual term will be much higher than that of other normal residual terms. This wrong residual term will concentrate computing resources, resulting in the lack of computing resources for the correct residual term. Incorrect residuals can be “intelligently” filtered using the Huber kernel [30]. When the error is greater than the threshold value, it will be judged as the wrong residual term, and the function of the first form must be used to limit the error growth. If the error is less than the threshold, the original form is kept, and the formula can be expressed as

$$H(e) = \begin{cases} \frac{1}{2}e^2 & \text{if } |e| \leq \delta \\ \delta(|e| - \frac{1}{2}\delta) & \text{otherwise} \end{cases} \quad (40)$$

### 3.3.4. Sliding Windows and Marginalization

In sum, it can be found that the optimization objective function of tightly coupled fusion data increases dynamically. With the continuous movement of the robot, the state variables in the objective function increase continuously, which brings difficulties to the optimization problem. In order to control the growth of optimization scale, this paper adopted the sliding window method. The sliding window will constantly discard the oldest state variables and add new ones, and the sum of state variables in the window will remain constant. Use of multiple state windows can make state estimation more accurate, and at the same time, a robust optimization cost function can be established to reduce the influence of external points on state estimation [31]. However, if we delete the old state directly while maintaining the sliding window, the original visual and IMU constraints will be broken. In this paper, the marginal method was used to transform the constraint information into the prior distribution of variables to be optimized.

Suppose that all states of a sliding window can be defined as follows

$$\Delta x = \begin{bmatrix} \Delta x_m \\ \Delta x_n \end{bmatrix} \quad (41)$$

where  $\Delta x_m$  and  $\Delta x_n$  represent  $m$  states to be removed and  $n$  states to be retained, respectively. Optimization problems can be expressed as

$$\begin{bmatrix} A & B \\ B^T & D \end{bmatrix} \begin{bmatrix} \Delta x_m \\ \Delta x_n \end{bmatrix} = \begin{bmatrix} g_m \\ g_n \end{bmatrix} \quad (42)$$

where  $ABD$  is the three matrix blocks of the  $H$  matrix, and  $g_m, g_n$  are the two vectors of the  $g$  vector.

There are a total of  $m + n$  states in the sliding window, and these  $m + n$  variables are solved as a whole. If the inverse operation of  $H$  matrix is solved directly, it will consume a large amount of computing resources. Moreover, the  $\Delta x_m$  needs to be deleted, so there is no need to perform a second calculation. Therefore, this paper chose to eliminate the  $B^T$  matrix block in the lower left corner, and the formula can be expressed as

$$\begin{bmatrix} A & B \\ 0 & D - B^T A^{-1} B \end{bmatrix} \begin{bmatrix} \Delta x_m \\ \Delta x_n \end{bmatrix} = \begin{bmatrix} g_m \\ g_n - B^T A^{-1} b_m \end{bmatrix} \quad (43)$$

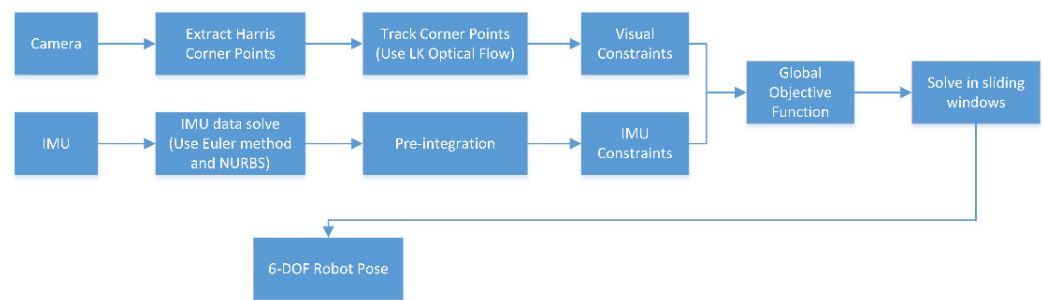
Thus, the second row of the equation becomes independent of  $\Delta x_m$ . The state variables to be retained in the sliding window can be obtained by directly solving the second line, and the formula can be expressed as

$$(D - B^T A^{-1} B) \Delta x_n = g_n - B^T A^{-1} b_m \quad (44)$$

Equation (41) requires only the inverse operation of the matrix block  $A$ . In addition, the number of deleted  $m$  variables is far less than the number of retained  $n$  variables, which greatly improves the computational efficiency.

In conclusion, there are three main purposes of marginalization: (1) the amount of computation is controlled by controlling the size of the sliding window; (2) ensuring that the frames in the sliding window have sufficient parallax; (3) the amount of operation is reduced by making the matrix dimension involved in the operation smaller.

In addition, the entire process of tightly coupled visual and IMU fusion can be described as in Figure 4.



**Figure 4.** The entire process of tightly coupled visual and IMU fusion.

#### 4. Experimental Results and Analysis

This chapter mainly carries on the experiment verification to the algorithm designed above. In this paper, the visual inertial navigation and positioning system were verified by dataset, quantitative positioning accuracy, and qualitative robustness experiments. Finally, the conclusion was drawn by analyzing the experimental results.

##### 4.1. Experimental on Dataset

This section uses datasets to test the performance of the proposed fusion location algorithm. KITTI and EUROC datasets are mainly used in this section. The image sequence selected by the KITTI dataset was high-speed dynamic pedestrians and vehicles, with nearly 5000 images, and the scene with the longest distance up to 400 m, simulating the scene of dynamic interference in the underground complex. The image sequence selected by the EUROC dataset is the factory of weak texture, with nearly 200 images, and the scene with the longest distance of 10 m, simulating the environment of weak texture in the underground complex. The positioning system designed in this paper and ORB-SLAM2 of pure visual positioning method were tested on these two data sets. Absolute pose error was used to evaluate the global consistency of the entire trajectory.

Figure 5 is the comparison of the GT (ground truth) of the KITTI dataset, the estimated value of the algorithm in this paper, and the estimated value of the pure visual method in six poses in space. The estimated data of X- and Z-axes were relatively close to the GT, while the data of the Y-axis had obvious deviation, the pitch angle was relatively close, and the roll angle and yaw angle had obvious deviation, but the overall trend of change was relatively consistent. As shown in Figure 6, by calculating APE, the maximum APE of ORB-SLAM2 was 7.48 m, while the maximum APE of the fusion positioning algorithm proposed in this paper was 2.76 m. It was shown that the fusion positioning system proposed in this paper can effectively reduce the errors caused by the pure visual positioning method under dynamic interference, and the improvement effect was up to 63.1%. In addition, the method adopted in this paper had lower variance and was more stable than ORB-SLAM2.

Figure 7 shows the comparison of the GT in the EUROC dataset, the estimated values of the algorithm in this paper, and the estimated values of the pure visual method in the six postures in space. Due to the serious interference of weak texture on the pure visual positioning method, the change trend of six pose was inconsistent with the GT. However, on the X-axis and Y-axis, the method used in this paper was closer to the GT than ORB-SLAM. As shown in Figure 8, by calculating APE, the maximum APE of ORB-SLAM2 was 0.238 m, while the maximum APE of the fusion positioning algorithm proposed in this paper was 0.155 m. It shows that the fusion positioning system proposed in this paper can effectively reduce the errors caused by the pure visual positioning method under the condition of weak texture, and the improvement effect was up to 34.9%.

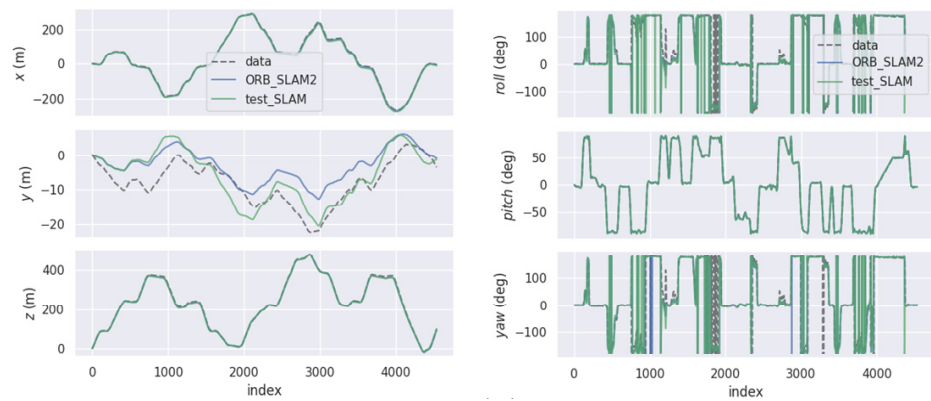


Figure 5. The comparison in space attitude of two algorithms in the KITTI dataset.

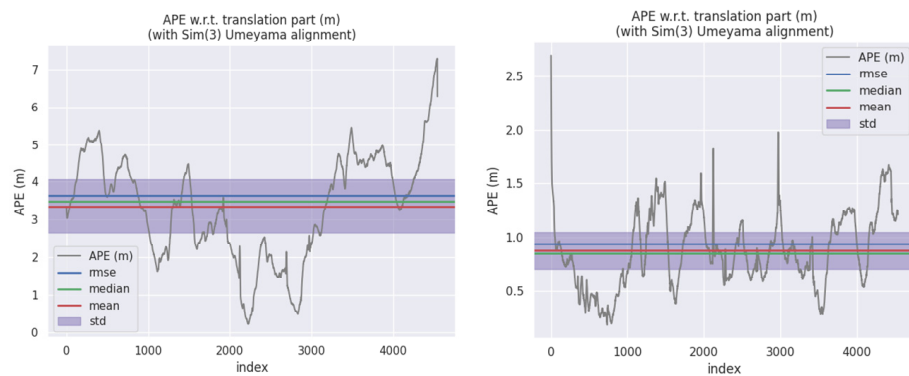


Figure 6. The comparison in APE of two algorithms in the KITTI dataset.

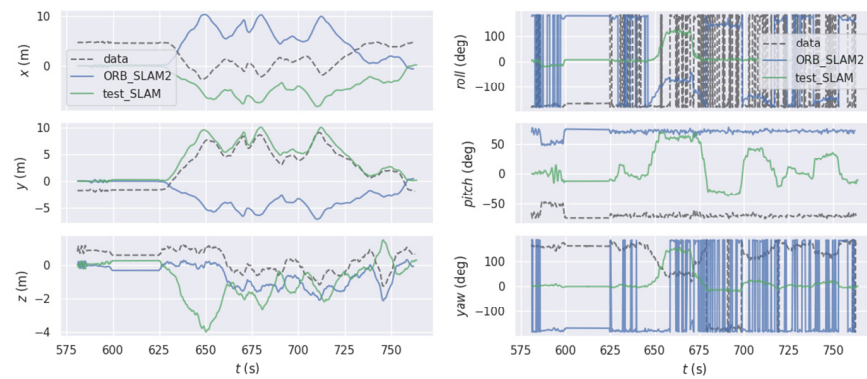


Figure 7. The comparison in space attitude of two algorithms in the EUROCC dataset.

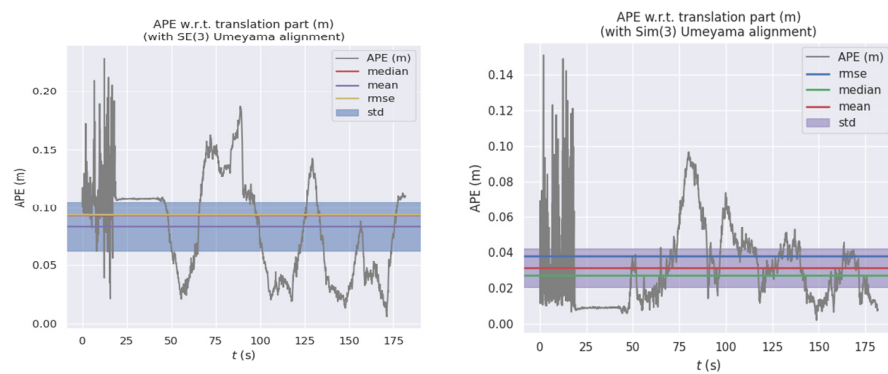


Figure 8. The comparison in APE of two algorithms in the EUROCC dataset.

#### 4.2. Experimental in Simulated Real Scene

In order to simulate dynamic interference and weak texture scenes of an underground complex, experiments in an underground complex involve public security and other issues. As shown in Figure 9, an indoor environment with weak texture was selected for the experiment, a number of dynamic personnel were added, and a rectangular experimental track of 4 m × 5 m was laid.

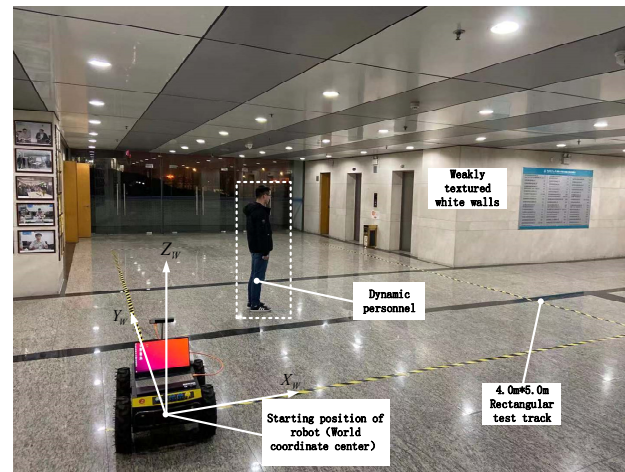


Figure 9. Deployment of the real environment.

##### 4.2.1. Positioning Accuracy Test

Due to the limitation of experimental conditions, the real pose value of the inspection robot at every moment cannot be obtained. In order to evaluate the positioning accuracy, this section conducts an evaluation from three perspectives: whether the estimated positioning trajectory was closed, the errors of the starting point and end point before and after fusion, and the trajectory deviation. A total of six groups of experiments were conducted. As shown in Figure 10, the left figure was the comparison of the positioning trajectory in the XOY plane after the pure inertial navigation and fusion. Due to the continuous accumulation of errors in the pure inertial navigation, the positioning curve was not closed, and the whole trajectory was seriously deformed. The trajectory estimated after fusion had smaller deviation and was closer to the ideal trajectory due to the addition of visual constraints. At the same time, vision can perform loop detection to make the final trajectory closed. The figure on the right shows the comparison between the pure inertial navigation and the fused positioning trajectory in the YOZ plane. The measurement results of pure inertial navigation diverged under the influence of other errors such as zero drift. The measured value of the fusion algorithm had some deviation and finally converged to a small error value.

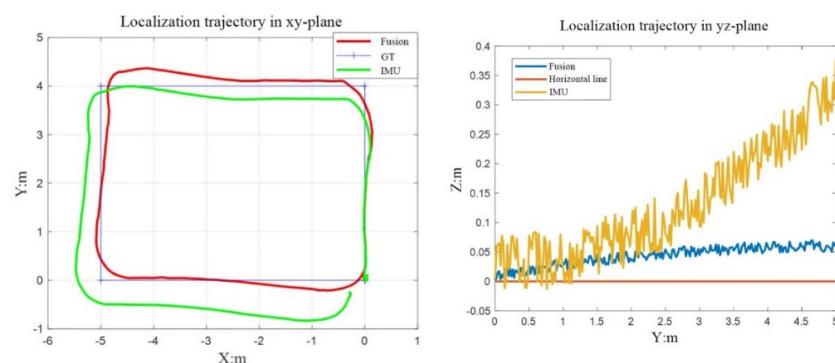


Figure 10. The comparison of localization accuracy between two algorithms in XOY and YOZ planes.

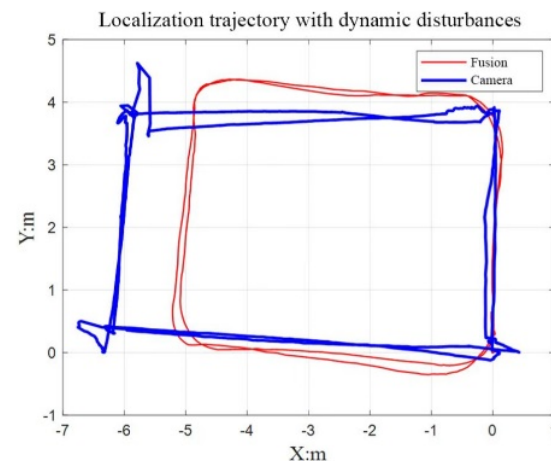
Furthermore, Table 2 shows the statistical data of the starting point and ending point error of six groups of experiments of the two algorithms before and after fusion. The error here was measured by the distance between the two points. The average error before fusion was 0.13 m, and the average error after fusion was 0.029 m. The overall experimental effect after fusion in this paper was improved by 77.7%.

**Table 2.** The comparison of error before and after sensor fusion.

Number	Pure Inertial Odometer			Fusion Algorithm Odometer		
	Origin	Destination	Error	Origin	Destination	Error
1	X: 0.01, Y: 0.04	X: −0.03, Y: −0.05	0.041	X: 0.03, Y: 0.04	X: 0.01, Y: 0.05	0.022
2	X: 0.02, Y: 0.01	X: −0.10, Y: −0.12	0.177	X: 0.02, Y: 0.04	X: 0.04, Y: 0.06	0.028
3	X: 0.05, Y: 0.03	X: −0.13, Y: −0.04	0.193	X: 0.07, Y: 0.02	X: 0.03, Y: 0.05	0.05
4	X: 0.03, Y: 0.02	X: −0.03, Y: −0.04	0.085	X: 0.06, Y: 0.02	X: 0.05, Y: 0.04	0.022
5	X: 0.04, Y: 0.01	X: −0.09, Y: −0.05	0.143	X: 0.02, Y: 0.04	X: 0.03, Y: 0.07	0.032
6	X: 0.02, Y: 0.05	X: −0.03, Y: −0.08	0.139	X: 0.05, Y: 0.04	X: 0.07, Y: 0.03	0.022
Average error		0.130			0.029	

#### 4.2.2. Robustness Experiments under Dynamic Disturbances

In this experiment, the fusion localization algorithm proposed in this paper was compared with the ORB-SLAM2 algorithm of binocular pure vision under dynamic interference. We had people try three types of exercise: running, continuous walking, and intermittent walking. In the experiments of running and continuous walking, there was no significant difference between the fusion localization algorithm and the pure visual localization method, and both of them had certain robustness to interference. However, in the experiment of intermittent walking, the difference between the two algorithms was particularly obvious. As shown in Figure 11, the inspection robot was made to complete the rectangular movement of  $4 \times 5$  m. At the same time, the experimenter continuously interfered with the binocular camera in the X-axis direction by intermittent walking (stopping for 20 s every 0.5 m).

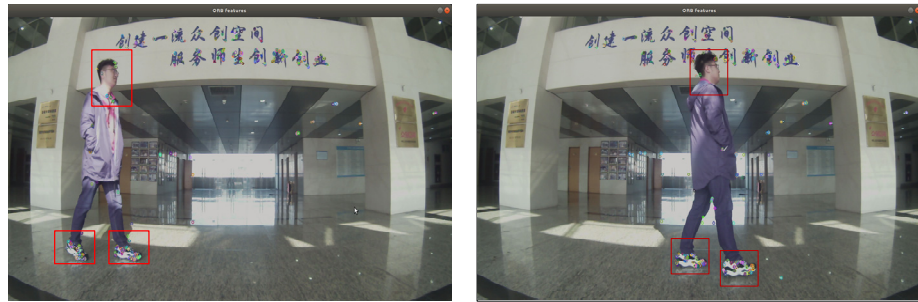


**Figure 11.** Track of two algorithms in dynamic interference.

From the experimental results, it can be found that the results of the pure visual positioning system were severely deformed, and the scale and ideal value were very different. However, the fusion localization algorithm proposed in this paper was affected by dynamic interference, and there was a certain degree of error in the localization scale, but the overall localization trajectory still maintained a smooth and flat rectangular trajectory. As shown in Figure 12, we used pure visual positioning of the ORB algorithm for image feature extraction, which can be found in the process of intermittent walking, and tester shoes and head were also full of the feature points; the ORB algorithm can extract all feature points in the environment, but not the difference between dynamic and non-rigid



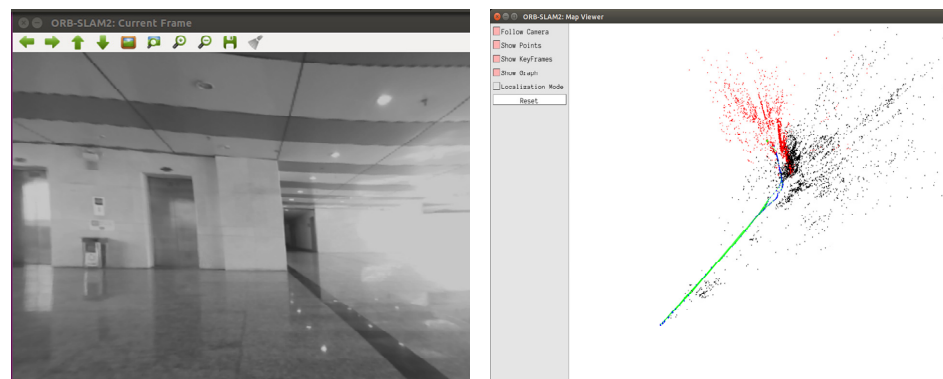
or static, rigid wall. When the visual sensor was still and the tester moved to another position, the visual sensor still misinterpreted the dynamic, non-rigid person as a static wall, resulting in a false displacement. As can be seen from the experimental results, the purely visual positioning method produced more displacement in the  $X$ -axis direction, and the own rotational motion of robot at the turning point led to more severe errors. The positioning method in this paper adopted the multi-sensor fusion method, and the inertial navigation measured the motion of the inspection robot itself, which was not affected by environmental changes. Therefore, the positioning trajectory was close to the ideal value, and the effect was better than the pure visual positioning method.



**Figure 12.** Location loss analysis.

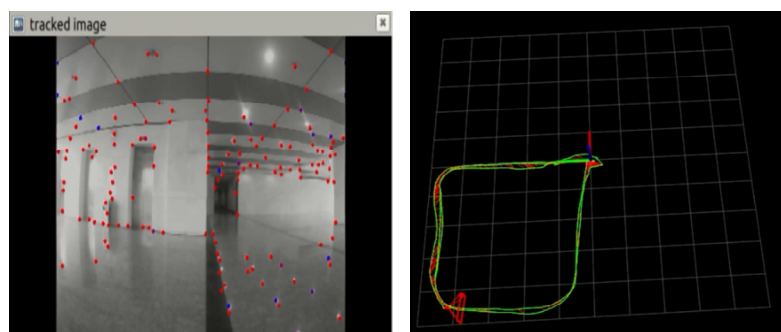
#### 4.2.3. Robustness Experiments with Missing Textures

In this experiment, the fusion localization algorithm proposed in this paper was compared with the ORB-SLAM2 algorithm of binocular pure vision in the absence of texture. The experimental environment simulated the underground complex, and the indoor environment with weak texture was selected for the experiment. As shown in Figure 13, when the inspection robot was displaced to the first corner, because the building facing smooth materials leveled off, the mirror was serious, and in some places, there was an overexposed photo; at the same time, in the face of the white wall texture feature not being obvious, on the basis of the characteristic point of the ORB-SLAM2 pure visual positioning method, it did not have enough ability to extract feature points in the space, and thus the whole algorithm failed to locate.



**Figure 13.** Localization results of ORB-SLAM2 in a low-texture environment.

As shown in Figure 14, the positioning method used in this paper relied not only on the currently collected image information but also on the image information before history in image processing, which is more robust than the feature point method relying on the current image each time. At the same time, due to the existence of inertial navigation, the algorithm can ensure the normal operation of the whole positioning system by measuring the own movement of the robot, even if there are some moments when the texture is missing.



**Figure 14.** Localization results of algorithm in the paper in a low-texture environment.

## 5. Conclusions

In this paper, the traditional visual SLAM algorithm was improved for the characteristics of large personnel flow and sparse background texture in an underground complex. In order to solve the problem that it is difficult for the front end to extract feature points from weak texture and the tracking is easy to be lost, the Harris algorithm with a stronger ability to extract feature points was introduced, which was combined with optical flow pyramid tracking to enhance the robustness of the visual positioning system. Various error sources of IMU were analyzed, and the performance of the Euler method and median method on dynamic model discretization was compared through simulation, and the Euler method with shorter time consumption was selected. Finally, the non-uniform rational B-spline was used to solve the problem of the non-synchronization of sensors with different frequencies, and the tight coupling between the vision sensor and IMU was realized. In the real physical environment, the research group simulated the environment of an urban underground complex, and the experiment showed that the fusion algorithm presented in this paper had better experimental results in terms of positioning accuracy, anti-dynamic interference, and robustness of weak texture than the positioning method using inertial navigation or vision only.

**Author Contributions:** K.Z. contributed to the conception of the study and manuscript preparation, J.C. performed the experiments, Y.J. and X.L. performed the data analyses, and J.C. and Z.Z. wrote the manuscript. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was supported by the National Key R&D Program of China under grant no. 2018YFB2101004.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Not applicable.

**Conflicts of Interest:** The authors declare no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

## References

1. Xu, X.Z.; Wang, J.C.; Zhang, L.X. *Physics of Frozen Soils*; Science Press: Beijing, China, 2001.
2. Zhang, J.; Singh, S. LOAM: Lidar odometry and mapping in real-time. *Robot. Sci. Syst.* **2014**, *2*, 1–9.
3. Hess, W.; Kohler, D.; Rapp, H.; Andor, D. Real-time loop closure in 2D LIDAR SLAM. In Proceedings of the 2016 IEEE International Conference on Robotics and Automation (ICRA), Stockholm, Sweden, 16–21 May 2016; pp. 1271–1278.
4. Shan, T.; Englot, B. Lego-loam: Lightweight and ground-optimized lidar odometry and mapping on variable terrain. In Proceedings of the 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Madrid, Spain, 1–5 October 2018; pp. 4758–4765.
5. Kim, G.; Kim, A. Scan context: Egocentric spatial descriptor for place recognition within 3d point cloud map. In Proceedings of the 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Madrid, Spain, 1–5 October 2018; pp. 4802–4809.

6. Shan, T.; Englot, B.; Meyers, D.; Wang, W.; Ratti, C.; Rus, D. Lio-sam: Tightly-coupled lidar inertial odometry via smoothing and mapping. In Proceedings of the 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Las Vegas, NV, USA, 24 October 2020–24 January 2021; pp. 5135–5142.
7. Engel, J.; Schöps, 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; pp. 834–849.
8. Wang, R.; Schworer, M.; Cremers, D. Stereo DSO: Large-scale direct sparse visual odometry with stereo cameras. *arXiv* **2017**, arXiv:1708.07878.
9. Forster, C.; Zhang, Z.; Gassner, M.; Werlberger, M.; Scaramuzza, D. SVO: Semidirect visual odometry for monocular and multicamera systems. *IEEE Trans. Robot.* **2016**, *33*, 249–265. [[CrossRef](#)]
10. Mur-Artal, R.; Montiel, J.M.M.; Tardos, J.D. ORB-SLAM: A versatile and accurate monocular SLAM system. *IEEE Trans. Robot.* **2015**, *31*, 1147–1163. [[CrossRef](#)]
11. Mur-Artal, R.; Tardós, J.D. Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras. *IEEE Trans. Robot.* **2017**, *33*, 1255–1262. [[CrossRef](#)]
12. Campos, C.; Elvira, R.; Rodríguez, J.J.G.; Montiel, J.M.M.; Tardós, J.D. ORB-SLAM3: An Accurate OpenSource Library for Visual, Visual-Inertial, and Multimap SLAM. *IEEE Trans. Robot.* **2021**, *2021*, 9440682. [[CrossRef](#)]
13. Sikka, P.; Asati, A.R.; Shekhar, C. Real time FPGA implementation of a high speed and area optimized Harris corner detection algorithm. *Microprocess. Microsyst.* **2021**, *80*, 103514. [[CrossRef](#)]
14. Jaiseeli, C.; Raajan, N.R. SLKOF: Subsampled Lucas-Kanade Optical Flow for Opto Kinetic Nystagmus detection. *J. Intell. Fuzzy Syst.* **2021**, *41*, 5265–5274. [[CrossRef](#)]
15. Příbyl, B.; Zemčík, P.; Čadík, M. Absolute pose estimation from line correspondences using direct linear transformation. *Comput. Vis. Image Underst.* **2017**, *161*, 130–144. [[CrossRef](#)]
16. Zheng, M.; Zhang, F.; Zhu, J.; Zuo, Z. A fast and accurate bundle adjustment method for very large-scale data. *Comput. Geosci.* **2020**, *142*, 104539. [[CrossRef](#)]
17. Tsintotas, K.A.; Bampis, L.; Gasteratos, A. Assigning visual words to places for loop closure detection. In Proceedings of the 2018 IEEE International Conference on Robotics and Automation (ICRA), Brisbane, QLD, Australia, 21–25 May 2018; pp. 5979–5985.
18. Gálvez-López, D.; Tardos, J.D. Bags of binary words for fast place recognition in image sequences. *IEEE Trans. Robot.* **2012**, *28*, 1188–1197. [[CrossRef](#)]
19. Galvez-Lopez, D.; Tardos, J.D. Real-time loop detection with bags of binary words. In Proceedings of the 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems, San Francisco, CA, USA, 25–30 September 2011; pp. 51–58.
20. Sivic, J.; Zisserman, A. Video Google: A text retrieval approach to object matching in videos. In Proceedings of the Ninth IEEE International Conference on Computer Vision, Nice, France, 13–16 October 2003; p. 1470.
21. Kourabaslou, S.S.; Zhang, A.; Atia, M.M. A Novel Design Framework for Tightly Coupled IMU/GNSS Sensor Fusion Using Inverse-Kinematics, Symbolic Engines, and Genetic Algorithms. *IEEE Sens. J.* **2019**, *19*, 11424–11436. [[CrossRef](#)]
22. Ling, Y.; Bao, L.; Jie, Z.; Zhu, F.; Li, Z.; Tang, S.; Liu, Y.; Liu, W.; Zhang, T. Modeling varying camera-imu time offset in optimization-based visual-inertial odometry. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 484–500.
23. Lovegrove, S.; Patron-Perez, A.; Sibley, G. Spline Fusion: A continuous-time representation for visual-inertial fusion with application to rolling shutter cameras. *BMVC* **2013**, *2*, 8.
24. Yang, Z.; Shen, S. Monocular visual-inertial state estimation with online initialization and camera-IMU extrinsic calibration. *IEEE Trans. Autom. Sci. Eng.* **2016**, *14*, 39–51. [[CrossRef](#)]
25. Forster, C.; Carlone, L.; Dellaert, F.; Scaramuzza, D. On-manifold preintegration for real-time visual-inertial odometry. *IEEE Trans. Robot.* **2016**, *33*, 1–21. [[CrossRef](#)]
26. Mur-Artal, R.; Tardós, J.D. Visual-inertial monocular SLAM with map reuse. *IEEE Robot. Autom. Lett.* **2017**, *2*, 796–803. [[CrossRef](#)]
27. Kaiser, J.; Martinelli, A.; Fontana, F.; Scaramuzza, D. Simultaneous state initialization and gyroscope bias calibration in visual inertial aided navigation. *IEEE Robot. Autom. Lett.* **2016**, *2*, 18–25. [[CrossRef](#)]
28. Nasiri, S.M.; Hosseini, R.; Moradi, H. A recursive least square method for 3D pose graph optimization problem. *arXiv* **2018**, arXiv:1806.00281.
29. Wu, Z.; Zhou, T.; Li, L.; Chen, L.; Ma, Y. A New Modified Efficient Levenberg–Marquardt Method for Solving Systems of Nonlinear Equations. *Math. Probl. Eng.* **2021**, *2021*, 5608195. [[CrossRef](#)]
30. Wang, H.; Li, H.; Zhang, W.; Zuo, J.; Wang, H. Derivative-free Huber–Kalman smoothing based on alternating minimization. *Signal Process.* **2019**, *163*, 115–122. [[CrossRef](#)]
31. Carlone, L.; Calafiore, G.C. Convex relaxations for pose graph optimization with outliers. *IEEE Robot. Autom. Lett.* **2018**, *3*, 1160–1167. [[CrossRef](#)]