sensors

MDPI

Article

A Coupled Visual and Inertial Measurement Units Method for Locating and Mapping in Coal Mine Tunnel

Daixian Zhu [1,*], Kangkang Ji [1,*], Dong Wu [1] and Shulin Liu [2]

[1] College of Communication and Information Engineering, Xi'an University of Science and Technology, Xi'an 710054, China
[2] College of Electrical and Control Engineering, Xi'an University of Science and Technology, Xi'an 710054, China
* Correspondence: zhudaixian@xust.edu.cn (D.Z.); 20207040032@stu.xust.edu.cn (K.J.)

Abstract: Mobile robots moving fast or in scenes with poor lighting conditions often cause the loss of visual feature tracking. In coal mine tunnels, the ground is often bumpy and the lighting is uneven. During the movement of the mobile robot in this scene, there will be violent bumps. The localization technology through visual features is greatly affected by the illumination and the speed of the camera movement. To solve the localization and mapping problem in an environment similar to underground coal mine tunnels, we improve a localization and mapping algorithm based on a monocular camera and an Inertial Measurement Unit (IMU). A feature-matching method that combines point and line features is designed to improve the robustness of the algorithm in the presence of degraded scene structure and insufficient illumination. The tightly coupled method is used to establish visual feature constraints and IMU pre-integration constraints. A keyframe nonlinear optimization algorithm based on sliding windows is used to accomplish state estimation. Extensive simulations and practical environment verification show that the improved simultaneous localization and mapping (SLAM) system with a monocular camera and IMU fusion can achieve accurate autonomous localization and map construction in scenes with insufficient light such as coal mine tunnels.

Keywords: visual feature tracking; sensor fusion; SLAM; tightly coupled

Citation: Zhu, D.; Ji, K.; Wu, D.; Liu, S. A Coupled Visual and Inertial Measurement Units Method for Locating and Mapping in Coal Mine Tunnel. *Sensors* **2022**, *22*, 7437. https://doi.org/10.3390/s22197437

Academic Editors: Yan Huang, Shiyang Tang, Zhanye Chen and Ping Guo

Received: 10 August 2022
Accepted: 19 September 2022
Published: 30 September 2022

1. Introduction

Simultaneous localization and mapping (SLAM) is an important technology for mobile robots to explore unknown areas [1]. The mainstream SLAM methods are divided into two types: vision and laser. A series of efficient, real-time pose-estimation methods have been proposed over the years. Laser SLAM is relatively mature in terms of theory, framework, and technology, and has matured applications in many aspects [2]. For example, the KUKA Navigation Solution in 2D-laser SLAM is used in sweeping robots. However, 2D SLAM can only obtain single plane information in the environment, which is not suitable for application in large complex scenes. In order to obtain more abundant environmental information, multi-line (Light Detection And Ranging) LiDAR or vision is usually used for positioning and mapping [3]. The increase in the number of radar lines means that more information can be obtained. At the same time, it also represents an increase in hardware costs. Vision sensors are cheaper compared to LiDAR. Maps constructed using vision sensors contain more information. Low power consumption and low weight make it possible to deploy vision sensors on mobile platforms with limited load. Vision SLAM has a wide range of applications both indoors and outdoors [4]. The image information obtained from the vision sensors is used to calculate the camera motion between adjacent images and thus estimate the camera motion trajectory. This method is called inter-frame estimation. As an important part of vision SLAM, the accuracy of inter-frame matching directly affects the results of localization and map building.

In recent years, many excellent visual SLAM algorithms have been proposed, and the real-time performance and positioning accuracy have been greatly improved. The method

of the characteristic point is most commonly used for inter-frame estimation. Feature points such as corner points, edge points, and blocks in adjacent images are extracted and used to estimate camera motion. Feature point-extraction algorithms include Scale Invariant Feature Transform (SIFT) [5], Speeded Up Robust Features (SURF) [6], Harris [7], Oriented Fast and Rotated Brief (ORB) [8], etc. In these feature-extraction methods, only the significant feature points in the image are extracted. Also to improve the real-time performance of the algorithm, the distribution of feature points is sparse. This makes the visual SLAM more sensitive to illumination and features are often lost under drastic changes in illumination and fast camera motion. For example, both ORB-SLAM2, proposed by Raul Mur-Artal et al., and DSO (Direct Sparse Odometry), proposed by Jakob et al. [9,10], localize by visual methods. They are able to achieve good localization accuracy when the ambient lighting conditions are good. However, they tend to perform poorly under fast motion and poor illumination conditions.

With the increasing requirements for localization accuracy and robustness in different application scenarios, it is not enough to collect information from a single sensor and then calculate the position. Therefore, more and more researchers are focusing on multi-source fusion approaches [11]. There are various ways of using multi-source fusion. Among them, the fusion of sensors is the most commonly used [12]. This includes fusion applications of two or more sensors in cameras, LiDAR, Inertial Measurement Unit (IMU), and GPS. There is also multi-feature fusion, which includes the extraction of features such as points, lines, and grayscale values to obtain multiple feature map elements.

Monocular visual SLAM can collect enough environmental information through cameras in indoor scenes with good lighting conditions and rich features. Then the camera state is estimated and the positional pose is calculated. However, when the camera moves rapidly, it is easy to lose point features with monocular visual SLAM [13]. Because of the limitations of camera frame rate and resolution, it is difficult to achieve accurate localization in fast motion. With its high acceleration and angular velocity measurement rates, the IMU can be used to assist vision sensors for feature tracking. Although the bias and noise of the IMU cause an accumulation of errors and drifts, they can be corrected by vision methods. The use of different sensors can complement each other to provide more accurate global or local positioning measurements for mobile robots. Many excellent SLAM algorithms for sensor fusion have been proposed in recent years. Methods that fuse vision and IMU, such as ORB-SLAM3 [14], use long-term data correlation to achieve high-precision localization in large scenes. Its fast and accurate IMU initialization and multi-session map merging capabilities enable robust operation in real-time in all kinds of scenes, but this comes at a higher computational cost. VI-DSO [15], proposed by Lukas et al., also balances speed and accuracy and enables real-time centimeter-level localization. However, this method requires a longer time for IMU initialization, even more than 30 s under poor lighting conditions. Based on the extended Kalman filter, the MSCKF [16] proposed by Mourikis et al. is a visual-inertial odometer that fuses visual and inertial information. Instead of estimating feature point states as system states, it uses a sliding window strategy to combine features from multiple observation frames. The filter is updated using the pose constraint between observation frames, which effectively improves the real-time performance of the algorithm. However, it has more error and insufficient localization accuracy when applied in large scenes without loopback detection. Leutenegger et al. proposed a nonlinear optimization-based visual-inertial navigation fusion method, OKVIS [17]. The scheme supports binocular cameras and uses a sliding window method to construct a minimum error function by visual reprojection constraints and IMU measurement constraints. OKVIS uses the First-Estimates Jacobian (FEJ) to ensure the consistency of the system. However, its slow computation speed is not suitable for scenes that require real-time feedback such as autonomous mobile robots. The VINS-Mono [18] proposed by Shen Shaojie et al. uses an optical flow method for motion tracking and pre-integrates IMU data.

Its fast initialization process and visual-inertial tightly coupled nonlinear optimized estimator achieve accurate pose estimation of the unmanned aircraft. Since the front-end

uses the optical flow method for tracking matching without extracting feature descriptors, feature points will be lost when the frame blurs due to violent motion. Moreover, it is more sensitive to light changes, which affects the positioning accuracy.

Because there are some limitations in all of the above methods, this paper proposes a localization and mapping method based on VINS-Mono. It achieves high accuracy and real-time mobile robot trajectory estimation and map construction through the coupling of vision and IMU. The method combines ORB features and line features to improve the accuracy of robot feature extraction in scenes with poor lighting conditions such as rugged ground and coal mine tunnels. It also reduces the feature loss due to motion blur and illumination changes. In addition, the method uses IMU data to assist visual features for localization and tracking, which provides more constraints on the estimation of the system state and improves the system localization accuracy. We experimented with the proposed method in real scenarios and EuRoC datasets [19]. The experimental results of the real Tunnel environment and the EuRoC dataset show that the improved algorithm can track stably in environments with fast motion and poor lighting conditions, and obtain higher accuracy of the positional estimation results.

The chapters of this article are organized as follows: Section 2 presents our work for the tightly coupled approach for the visual-inertial SLAM algorithm. The relevant experiments and results are analyzed in Section 3, followed by conclusions in Section 4.

2. Materials and Methods

The system block diagram of the positioning and mapping algorithm based on the fusion of visual features and IMU proposed is shown in Figure 1. The positioning and mapping algorithm for the fusion of visual features and IMU is mainly divided into front-end, system initialization, and back-end.

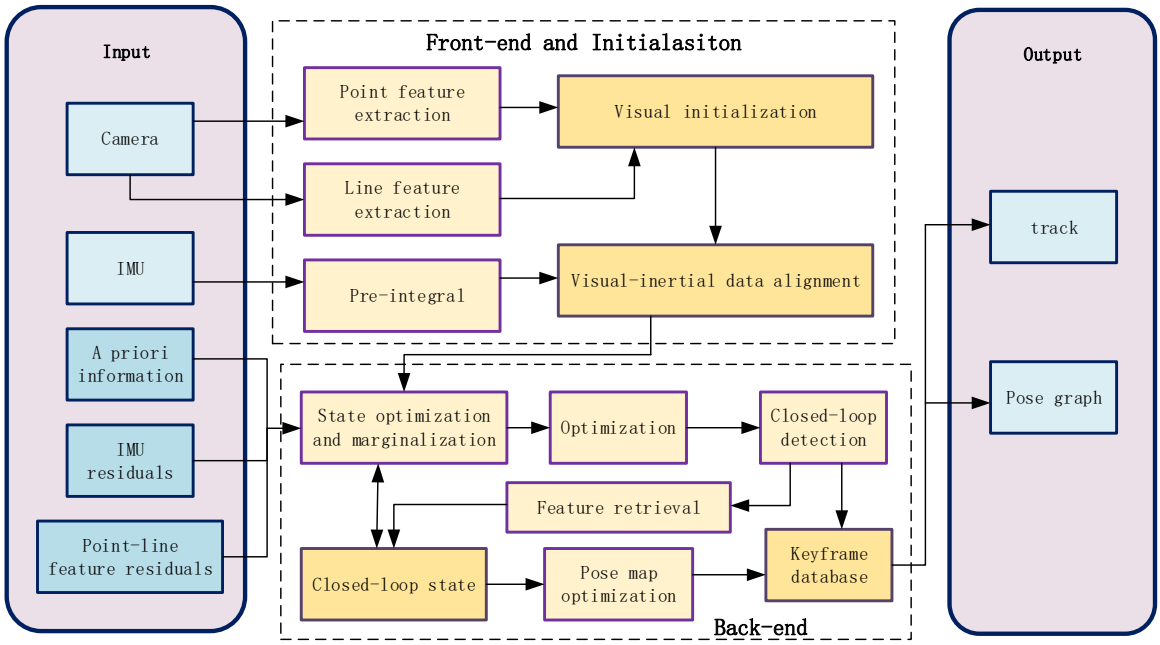


Figure 1. System block diagram.

2.1. Front-End Data Association

In visual SLAM, the front-end processing is to extract the feature points of adjacent frames from the image information collected by the camera, and then perform feature-point matching to calculate the correlation information between image frames. The pre-integration of the IMU information is used as a constraint between image frames [20]. The back-end recovers the depth information from the input-associated information and generates a pose graph. Feature maps are constructed with feature points for closed-

loop detection and subsequent robot repositioning. In SLAM, the front-end processing requires not only the accuracy of feature-point matching but also real-time calculation. By comparing the advantages and disadvantages of various feature-point extraction methods, ORB features are finally selected as the feature-extraction object of this system to meet the accuracy and real-time performance of feature extraction and matching [21]. In addition, compared with point features, line features provide more information about the geometric structure of the environment, and have better robustness to illumination changes and texture sparsity.

2.1.1. Extraction and Matching of Image Feature Points

The ORB feature is a highly matching computationally efficient feature that makes the features from accelerated segment test (FAST) directional and uses the binary descriptor BRIEF to describe them [22–24].

The FAST feature points do not have rotation invariance, so it is necessary to add a direction description to each feature point by the intensity centroid method. The implementation steps of the algorithm are as follows. First, the moment of the image block is defined in the image block as mentioned in [22], and shown in the following formula:

$$m_{pq} = \sum_{x,y \in B} x^p y^q I(x,y) \quad p,q \in \{0,1\} \tag{1}$$

where $I(x, y)$ is the gray value of the pixel. The centroid is expressed as:

$$\mathrm{C} = \left(\frac{m_{10}}{m_{00}}, \frac{m_{01}}{m_{00}} \right) \tag{2}$$

the direction of the feature point is the line connecting the geometric center O and the centroid C of the image block, and the direction vector is $\overrightarrow{\mathrm{OC}}$.

The FAST feature points have scale invariance and rotation invariance through the image pyramid and gray centroid method. BRIEF is a binary descriptor and uses Hamming distance to do matching on feature points. The XOR operation greatly improves the computational efficiency and makes the matching of ORB features have an inherently real-time capability.

2.1.2. Extraction and Matching of Image Feature Lines

The extraction of feature lines is based on the Line Segment Detector (LSD) [25] straight-line detection algorithm. This achieves sub-pixel accuracy of line segment detection results and controls the number of false detections.

The LSD algorithm uses the image pyramid method to perform Gaussian blurring on the original image to construct a scale space. The direction in which the image pixels change rapidly is the image-gradient direction, and the contour-line direction is perpendicular to the gradient direction. The area composed of pixels with approximately the same gradient direction in the image of the same scale is called the line-support region.

The description of the line segment adopts the LBD [26] descriptor, which defines dL as the line-segment direction, and $d_\perp$ as the line-segment orthogonal direction. The construct strip descriptor BD_j is computed from the strips B_j, B_{j-1}, and B_{j+1} in the line support domain.

The expression for the k row of the strip is:

$$\begin{cases} v1_j^k = \lambda \sum_{gd_\perp > 0} gd_\perp & v2_j^k = \lambda \sum_{gd_\perp < 0} gd_\perp \\ v3_j^k = \lambda \sum_{gd_\mathrm{L} > 0} gd_\perp & v4_j^k = \lambda \sum_{gd_\mathrm{L} < 0} gd_\perp \end{cases} \tag{3}$$

$$\lambda = f_g(k) f_l(k) \tag{4}$$

where g represents the image pixel gradient value. $gd_\perp$ and gd_L represent the projection of gradient values in two directions. λ is the Gaussian coefficient; $fg(k)$ is the global weighted

Gaussian function, which reduces the influence of the gradient of pixels farther from the edge of the line segment on the descriptor; $fl(k)$ is the local weighted Gaussian function to reduce the boundary effect which comes from the descriptor transitions from one region to another.

The LBD descriptor matrix corresponding to each BD_j is:

$$BDM_j = \begin{bmatrix} v1_j^1 & \cdots & v1_j^n \\ \vdots & \ddots & \vdots \\ v4_j^k & \cdots & v4_j^n \end{bmatrix} \tag{5}$$

BD_j consists of the mean direction M_j of BDM_j and the standard deviation vector S_j:

$$\text{LBD} = \left(M_1^T, S_1^T, \ldots, M_m^T, S_m^T\right)^T \tag{6}$$

Line-feature matching is achieved by judging the Hamming distance of the LBD descriptor. The matching results with too short line segments and too large included angles are eliminated.

2.1.3. IMU Pre-Integration

In general, the sampling frequency of the camera is 30 Hz, whereas the sampling frequency of the IMU can reach up to 1000 Hz. The keyframe mechanism is often used to guarantee the real-time nature of SLAM. As shown in Figure 2, there are many IMU data between adjacent keyframes, and the motion information between keyframes can be calculated by the method of multi-view geometry [27]. The IMU integral term in the world coordinate system contains the rotation matrix *Rbwk* of the IMU coordinate system relative to the world coordinate system. During the optimization process, the change of the key frame bit pose causes the corresponding *Rbwk* to change as well. Then it is necessary to repeat the integration, leading to a large increase in computation. The pre-integration means that the reference coordinate system for IMU integration is converted to the body coordinate system of the previous frame, then the motion between the two frames is calculated to avoid double integration.

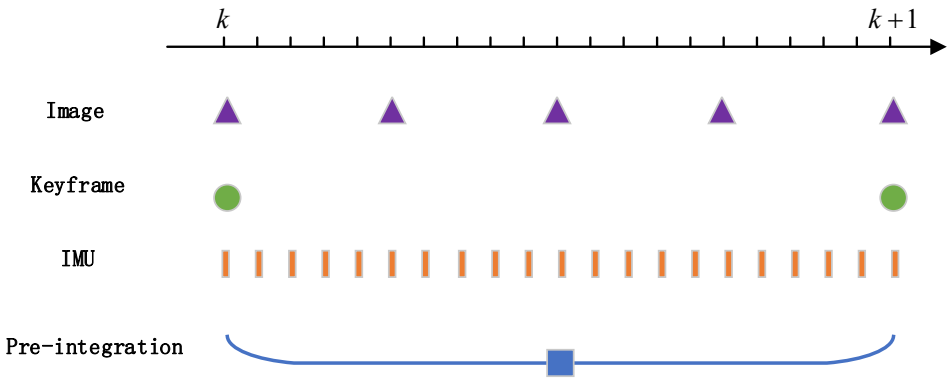


Figure 2. IMU pre-integration.

The moments t_k and t_{k+1} correspond to two consecutive keyframes x_k and x_{k+1}. The position $p_{b_k}^w$, velocity $v_{b_k}^w$, and rotation state $q_{b_k}^w$ of the system are propagated based on IMU measurements in the interval $[t_k,t_{k+1}]$. b_k represents the IMU body coordinate system at time k, and w represents the world coordinate system.

$$\begin{cases} p_{b_{k+1}}^w = p_{b_k}^w + v_{b_k}^w \Delta t_k + \iint_{t\epsilon[t_k,t_{k+1}]} (R_t^w(a_t - b_{a_t} - n_a) - g^w)dt^2 \\ v_{b_{k+1}}^w = v_{b_k}^w + \int_{t\epsilon[t_k,t_{k+1}]} (R_t^w(a_t - b_{a_t} - n_a) - g^w)dt \\ q_{b_{k+1}}^w = q_{b_k}^w \otimes \int_{t\epsilon[t_k,t_{k+1}]} \frac{1}{2} q_t^{b_k} \otimes (w_t - b_{w_t} - n_w)dt \end{cases} \tag{7}$$

The propagation of states such as system position, velocity, and rotation depends on the position $p^w_{b_k}$, velocity $v^w_{b_k}$, and rotation $q^w_{b_k}$ at keyframe x_k moments. When these initial states change after optimization, the state propagation needs to be repeated. This will waste a lot of computing resources. The left side of Equation (8) should be multiplied by $R^w_{b_k}$, and the reference coordinate system from the world coordinate system adjusted to the body coordinate system of the keyframe at k time. This will achieve the relative motion delta independent of the state of x_k:

$$\begin{cases} R^{b_k}_w p^w_{b_{k+1}} = R^{b_k}_w (p^w_{b_k} + v^w_{b_k} \Delta t_k - \frac{1}{2} g^w \Delta t_k^2) + \alpha^{b_k}_{b_{k+1}} \\ R^{b_k}_w v^w_{b_{k+1}} = R^{b_k}_w (v^w_{b_k} - g^w \Delta t_k) + \beta^{b_k}_{b_{k+1}} \\ q^{b_k}_w \otimes q^w_{b_{k+1}} = \gamma^{b_k}_{b_{k+1}} \end{cases} \tag{8}$$

$$\begin{cases} \alpha^{b_k}_{b_{k+1}} = \iint_{t \epsilon [t_k, t_{k+1}]} \left(R^{b_k}_t (a_t - b_{a_t} - n_a) \right) dt^2 \\ \beta^{b_k}_{b_{k+1}} = \int_{t \epsilon [t_k, t_{k+1}]} (R^w_t (a_t - b_{a_t} - n_a) - g^w) dt \\ \gamma^{b_k}_{b_{k+1}} = \int_{t \epsilon [t_k, t_{k+1}]} \frac{1}{2} \Omega (w_t - b_{w_t} - n_w) \gamma^{b_k}_t dt \end{cases} \tag{9}$$

The reference coordinate system in the integral term is the body coordinate system of the frame k. The integral term for the frame $k+1$ is only related to $\boldsymbol{a_t}$ and $\boldsymbol{w_t}$ at different moments. Even if the state of the key frames, such as position, velocity, and rotation is adjusted during optimization, it will not have any effect on the integral term and avoids repeated integration.

2.2. System Initialization

By extracting and matching point and line features, the association between pixels of adjacent frames can be established. Through visual initialization, the pose in the three-dimensional space of the continuous frame camera and the landmark composed of points and lines can be obtained. Assuming that the image frame to be initialized in the sliding window is shown in Figure 3, the goal of visual initialization is to calculate the landmark and the pose of the camera in the sliding window through the structure from motion (SFM) [28].

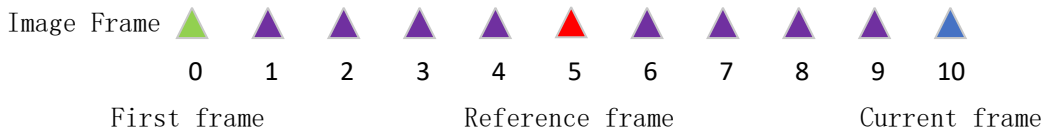


Figure 3. Image frame initialization.

The reference frame is selected by taking the keyframe with more feature matches in the sliding window as the reference frame. The reference frame and the current frame are polar-constrained and triangulated to calculate the bit pose from the current frame to the reference frame and the 3D roadmap of the co-view. For a frame between the reference frame and the current frame, Perspective-n-points (PnP) [14] can be used to calculate the bit pose of that frame to the current frame, then to triangulate the waymark points of that frame to the current frame. The same operation is performed for a frame between the first frame and the reference frame. Finally, the other unrecovered waypoints are trigonometric again, and all the bit-postures and waypoints in the obtained sliding window are optimized to complete the visual initialization.

Finally, the trajectories calculated by the vision and the IMU are aligned. Then the gyroscope bias, initial velocity, gravity, and scale factor can be assign at the start moment. This completes the system initialization process.

2.3. The Back-End Fusion Method

Through the extraction and matching of point-line features and visual initialization, pure vision can resolve adjacent frame poses and co-visualized 3D landmarks. The reprojection residuals are established as the association between image frames for optimizing the poses. The IMU data are pre-integrated as the key inter-frame constraint and also used to establish the image inter-frame constraint [29]. Finally, the optimization based on sliding windows is performed to estimate the optimal poses.

2.3.1. Visual Feature Point and Feature Line Reprojection Model

The visual feature point reprojection residual describes the distance between the projection point and the observation point of the landmark in the three-dimensional space under the normalized camera coordinate system.

As shown in Figure 4, the visual observations of map point P on the plane of reference frame i are converted to the pixel coordinates of the current j camera coordinate system, and the defined reprojection residual term is:

$$r_f\left(\hat{z}_{f_k}^{C_i}, \chi\right) = \begin{bmatrix} \frac{x^{c_j}}{z^{c_j}} - u_{f_k}^{c_j} \\ \frac{y^{c_j}}{z^{c_j}} - v_{f_k}^{c_j} \end{bmatrix} \tag{10}$$

$$\begin{bmatrix} x^{c_j} \\ y^{c_j} \\ z^{c_j} \end{bmatrix} = \boldsymbol{T}_b^c \boldsymbol{T}_w^{b_j} \boldsymbol{T}_{b_i}^w \boldsymbol{T}_c^b \frac{1}{\lambda_k} \begin{bmatrix} u_{f_k}^{c_i} \\ v_{f_k}^{c_i} \\ 1 \end{bmatrix} \tag{11}$$

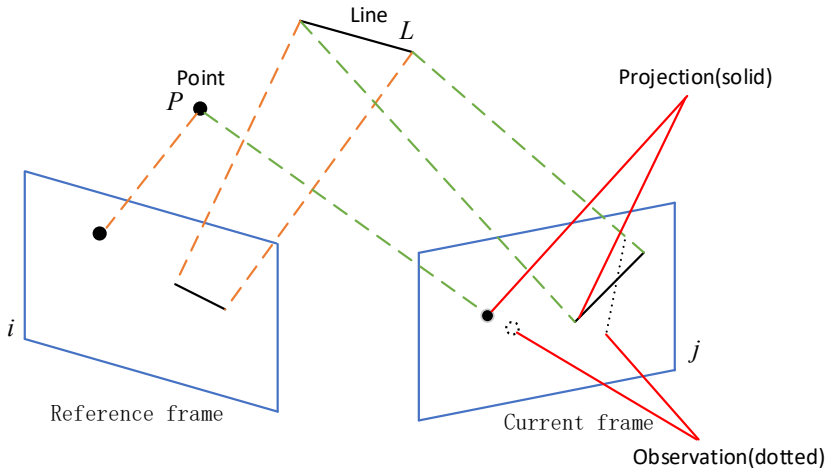


Figure 4. Reprojection error of point and line features.

The reprojection error of the feature line describes the distance between the projected position of the two endpoints of the line on the normalized image coordinate system and the predicted position obtained by inter-frame transformation. The reprojection error can be expressed as:

$$r_L\left(\hat{z}_l^{C_j}, \chi\right) = \begin{bmatrix} d_1 \\ d_2 \end{bmatrix} = \begin{bmatrix} \frac{p_s'^T l'}{\sqrt{l_1^2 + l_2^2}} \\ \frac{p_e'^T l'}{\sqrt{l_1^2 + l_2^2}} \end{bmatrix} \tag{12}$$

where L is a straight line in space. Its projected position is expressed as $l = [l_1, l_2, l_3]$, and its endpoints are ps and pe. The predicted position l', endpoints ps' and pe' are obtained by inter-frame transformation.

According to the visual point reprojection formula and the line reprojection formula, the Jacobian matrix relative to each optimization variable is obtained for back-end optimization.

2.3.2. IMU Pre-Integration Residual Model

The IMU pre-integration residual describes the difference between the measured value calculated by IMU pre-integration and the estimated value calculated by pose estimation. The optimized object is the pose of the keyframes. Assuming that the pose of keyframe x_1 is $\boldsymbol{T_1}$, and the pose of keyframe x_2 is $\boldsymbol{T_2}$, the relative poses of these two keyframes are:

$$\boldsymbol{T_{21}} = \boldsymbol{T}_1 * \boldsymbol{T}_2^{-1} \tag{13}$$

$\boldsymbol{T_{21}}$ is the estimation item, which is directly calculated from the camera pose. IMU pre-integration is equivalent to a measure between keyframes. Then the error term can be obtained:

$$r_{21} = \boldsymbol{T}_1 * \boldsymbol{T}_{imu}^{-1} \tag{14}$$

The keyframe pose is optimized by nonlinear optimization of the error term function in [30]. The residual term of the IMU can be expressed as:

$$r_B\left(z_{b_k b_{k+1}}, \chi\right) = \begin{bmatrix} r_p \\ r_q \\ r_v \\ r_{b_a} \\ r_{b_g} \end{bmatrix} = \begin{bmatrix} \boldsymbol{q}_{\omega b_i}^*(\boldsymbol{p}_{\omega b_j} - \boldsymbol{p}_{\omega b_i} - \boldsymbol{v}_i^{\omega}\boldsymbol{\Delta t} + \frac{1}{2}\boldsymbol{g}^{\omega}\boldsymbol{\Delta t}^2 +) - \boldsymbol{\alpha}_{b_i b_j} \\ 2[\boldsymbol{q}_{b_i b_j}^* \otimes (\boldsymbol{q}_{\omega b_i}^* \otimes \boldsymbol{q}_{\omega b_i})]_{xyz} \\ \boldsymbol{q}_{\omega b_i}^*\left(\boldsymbol{v}_j^{\omega} - \boldsymbol{v}_i^{\omega} + \boldsymbol{g}^{\omega} \triangle \boldsymbol{t}\right) - \boldsymbol{\beta}_{b_i b_j} \\ \boldsymbol{b}_j^a - \boldsymbol{b}_i^a \\ \boldsymbol{b}_j^g - \boldsymbol{b}_i^g \end{bmatrix} \tag{15}$$

2.3.3. Graph Optimization Model

Compared with the loosely coupled method, the tightly coupled method can obtain higher trajectory accuracy [10,11]. The state variables in the tightly coupled approach will become continually larger in size with the operation of the algorithm. In order to limit the computational cost, a graph optimization method based on sliding windows [31] is adopted. The state quantity to be optimized in the sliding window is:

$$\chi = \left[x_1, x_2, x_3, \ldots, x_N, \lambda_1, \lambda_2, \lambda_3, \ldots, \lambda_M, \mathrm{O}_1, , \mathrm{O}_2, \mathrm{O}_3, \ldots, \mathrm{O}_O, T_c^b\right] \tag{16}$$

$$x_k = \left[p_{wb_k}, q_{wb_k}, v_k^w, b_a^{b_k}, b_g^{b_k}\right]^T, k \in [1, N] \tag{17}$$

x_k is the status of the camera in the sliding window; p_{wbk} is the position of the camera; q_{wbk} is the orientation of the camera; v_k^w is the speed of the camera; $b_a^{b_k}$ is the bias of the accelerometer; $b_g^{b_k}$ is the bias of the gyroscope; λ_k is the inverse depth of the 3D point; O_k is the orthogonal expression of the feature line; T_c^b is the external parameter from the camera to the IMU; N is the number of keyframes in the sliding window; M and O are the number of point marks and line marks observed in all keyframes in the sliding window, respectively. As the algorithm runs, the oldest keyframe state variables in the sliding window need to be removed, and new keyframe state variables need to be added. The culled oldest frame state variables contain a lot of prior information.

The algorithm for graph optimization based on sliding windows not only uses the IMU data for state prediction but also IMU data as measurement information to optimize X. The optimal estimate of state X can be obtained by minimizing the residual of the state quantity within the sliding window, and its specific form is:

$$\begin{aligned} \chi_{MLE} = argmin \quad \Bigg(& \|r_p - J_p\chi\|^2 + \sum_{(i,j)\epsilon K} \rho\left(\left\|r_f\left(\hat{Z}_{f_k}^{C_i}, \chi\right)\right\|^2\right) + \sum_{k\epsilon P} \rho\left(\left\|r_B\left(z_{b_k b_{k+1}}, \chi\right)\right\|^2\right) \\ & + \sum_{(i,j)\epsilon K} \rho\left(\left\|r_L\left(z_{L_j}^{c_k}, \chi\right)\right\|^2\right)\Bigg) \end{aligned} \tag{18}$$

$\rho(\cdot)$ is a robust kernel function to suppress false matching of outliers; r_p is the prior residual; r_c represents the reprojection residual of visual point features; r_L represents the reprojected residual of visual line features; r_B represents the reprojection residual of the visual line features of the IMU pre-integration residuals; $z_{L_j}^{c_k}$ is the observation at time k relative to road sign j; $z_{b_k b_{k+1}}$ is the observed value of pose transformation obtained through IMU data at time k and time $k+1$. As shown in Figure 5, in order to optimize the constraints between variables, the points and lines in the image that have a co-view relationship are associated with poses through visual constraints. Motion information between image keyframes is associated with IMU pre-integration constraints.

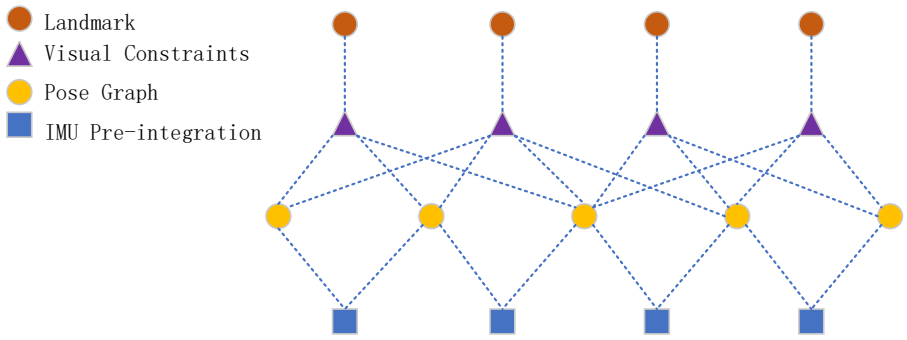


Figure 5. Data association.

2.4. Closed-Loop Detection and Global Pose Optimization

The system error is effectively constrained by the optimization based on the sliding window, but the accumulated error still exists. For the accumulated error, the closed-loop detection is used to determine whether the historical position has ever been reached. If successful detection of a return to the historical position is achieved, a closed-loop constraint is formed. When sufficient closed-loop constraints are obtained, they can be used for the optimization of global poses to reduce the cumulative error.

The detection of closed-loop in visual SLAM is to find image similarity. A certain class of vectors with similar features in an image is called a word, and a dictionary is a collection of visual words. The closed-loop detection based on the Bag of Words (BoW) model [32] is effective. The feature points in the image are described using the Bag of Words model to determine which word the feature points belong to, and the set of words of different classes of feature points in the image constitutes a dictionary. The feature points in the current key frame image and the dictionary elements are compared by the bag of words to determine whether the images are similar and whether closed-loop is detected. The candidate frame with the highest similarity to the current frame is detected; feature matching and geometric verification is performed; and when the matching feature points reach a sufficient number, it is considered to constitute a closed loop and global bit pose optimization is performed, otherwise, the image words are added to the dictionary.

The global pose optimization is to perform pose adjustment when a closed-loop is detected. As shown in Figure 6, the pose of the current frame is T_j, and it is detected that the i frame in the history frame is a closed-loop pose of T_i. The relative pose of the i frame and the j frame is T_{ij}. Without accumulated error, T_{ij} is strictly equal to ${T_i}^{-1}T_j$, but due to accumulated error, the residual can be constructed as shown in Equation (19). Adjust the pose to minimize the residual term to eliminate accumulated errors.

$$e_{ij} = ln\left({T_{ij}}^{-1}{T_i}^{-1}T_j\right)^{\vee} \tag{19}$$

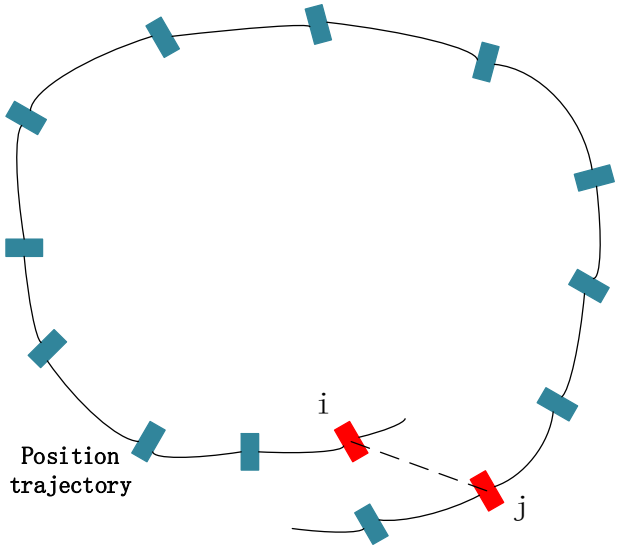


Figure 6. Closed-loop detection.

3. Results and Discussion

The EuRoC dataset was used to simulate the front-end vision processing part and the SLAM algorithm based on a monocular camera and IMU fusion respectively. It was obtained by using drone-mounted image sensors and IMU to collect data from 30 m^2 ordinary rooms and 300 m^2 factory environments. The dataset provided binocular camera images at 20 Hz and IMU measurements at 200 Hz. The pose transformation and trajectory of the aircraft during motion were obtained through the millimeter-level motion tracking system, and were used as the true values for the verification of the algorithm below.

In the indoor scene experiment, the algorithm in this paper was deployed on the Manifold-2C computing platform [33], which combined image data collected by visual sensors and IMU data to perform pose calculations in the laboratory scene.

Images with drastic changes in illumination, sparse texture, and poor illumination in the sequence were selected to test the visual front-end. Feature points and feature lines were extracted to produce accuracy statistics. The test for the overall positioning ability of the algorithm was to use the algorithm to estimate the poses of eleven sequences and to compare and evaluate these with the true value. Finally, the results were analyzed to verify the reliability and validity of our improved algorithm under different conditions.

3.1. Feature-Matching Evaluation

In the V1_03_difficult sequence, there were image frames with dramatic lighting changes and few textures. Two consecutive frames with large exposure differences in the sequence were selected to evaluate the robustness of the front-end feature extraction and matching algorithm.

The feature points and feature lines of the two sequences of images were extracted and matched. The results are shown in Figures 7 and 8.

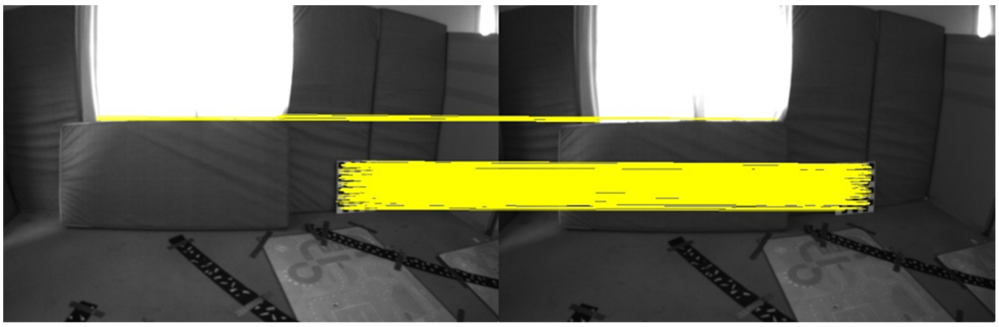


Figure 7. Results of feature point extraction and matching.

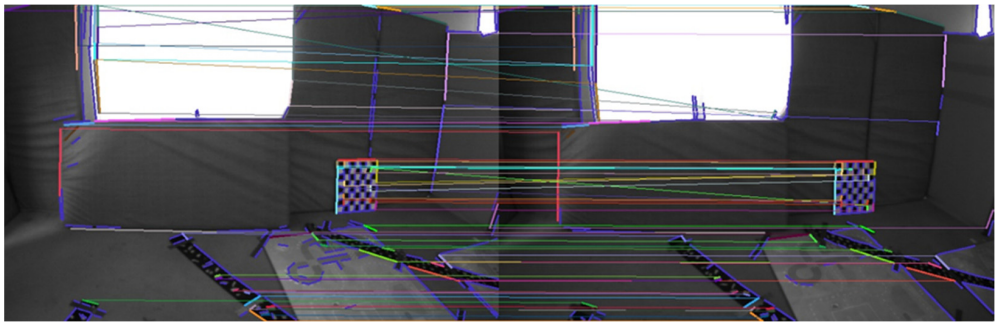


Figure 8. Results of feature line extraction and matching.

There are darker image frames with insufficient lighting in the MH_05_difficult sequence. The feature points and feature lines of the two sequences of images, respectively, were extracted and matched. The results are shown in Figures 9 and 10.

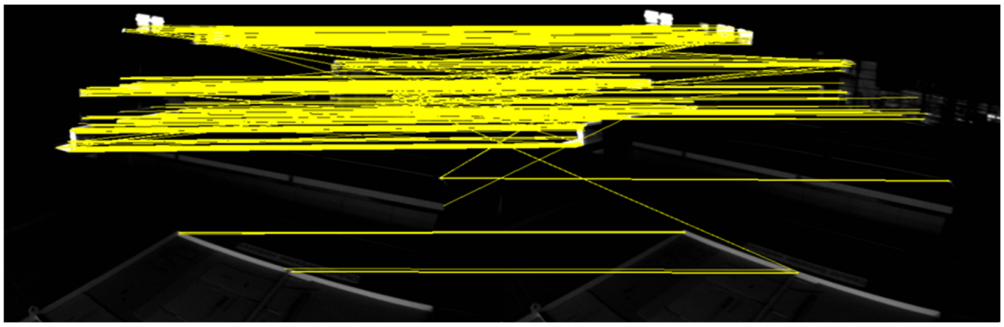


Figure 9. Image with insufficient illumination.

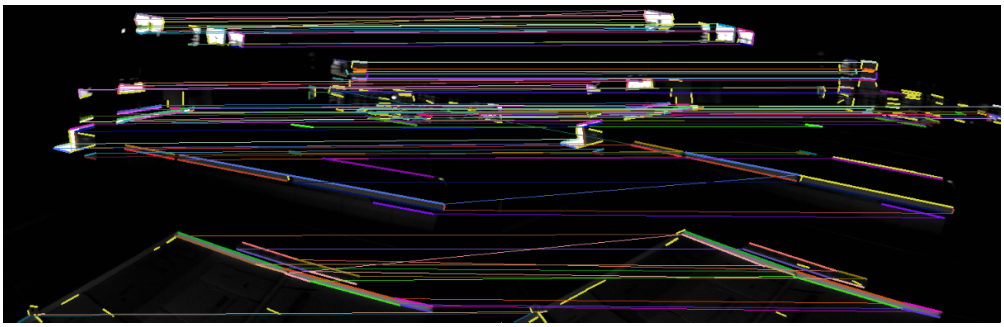


Figure 10. Image with insufficient illumination.

The image frames selected in the V1_03_difficult sequence had drastic changes in illumination and sparse texture, and the matching pairs of feature points were mainly

concentrated in the regions with obvious features. The image frames selected in the MH_05_difficult sequence were due to insufficient lighting, and the matching pairs of feature points were concentrated in local areas and had obvious mismatches. The matching pairs of feature lines were evenly distributed in the scenes with drastic changes in illumination and sparse textures, or scenes with insufficient lighting and relatively dark scenes. It can be seen that the geometric line feature had better robustness in the case of poor lighting. The matching accuracy rates of different features are shown in Table 1.

Table 1. Result of feature extraction and matching.

Feature Type	V1_03_Difficult Correct Rate	V1_05_Difficult Correct Rate
Point	79.4%	68.6%
Line	**86.7%**	**89.7%**

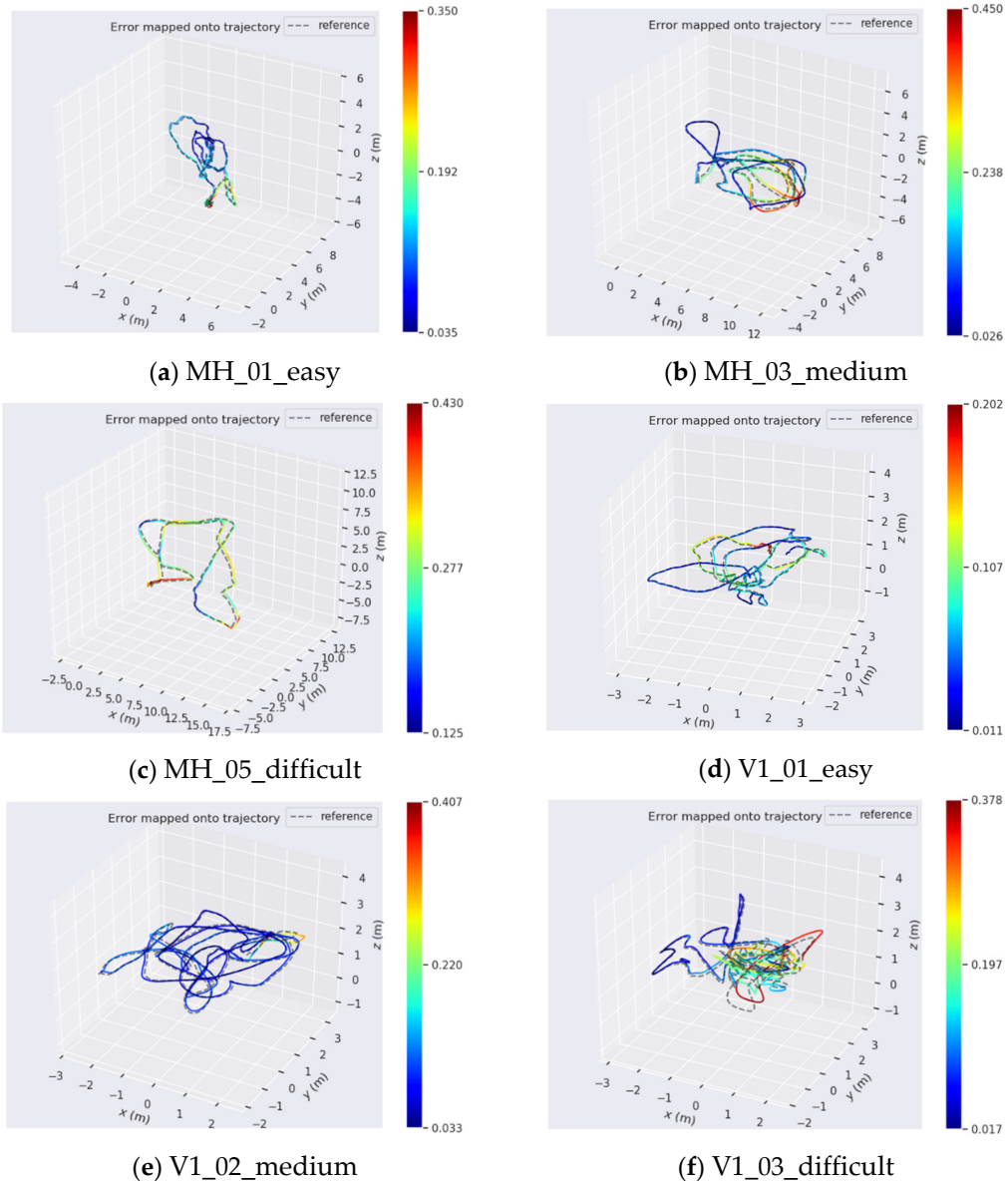
The correct rate of feature-point matching is significantly lower than that of feature line matching. The extraction and matching of feature lines have better robustness to scenes with changing illumination. Using feature points combined with feature lines in the visual front-end can effectively improve the robustness in scenes with severe illumination changes and avoid feature loss.

3.2. Odometer Accuracy Evaluation

Taking the MH_02 data sequence as an example, a total of 3040 sets of image information and 30,400 IMU information were provided. This algorithm used a monocular camera and selected the image information of cam0 as the input image data. The size of the image was an 8-bit grayscale image of 752 × 480 pixels. Some input images are shown in Figure 11. The IMU data formats in the dataset are timestamp, 3D angular velocity vector, and 3D acceleration vector. The final evaluation standard adopts the absolute pose error (APE) and compares it with the VINS-Mono algorithm.

Figure 11. Partial dataset image.

Figure 12 shows the trajectory error results of the 6 sequence ground-truth trajectories (dotted lines) in the EuRoc dataset and the trajectories (solid lines) obtained by our algorithm.



(**a**) MH_01_easy (**b**) MH_03_medium

(**c**) MH_05_difficult (**d**) V1_01_easy

(**e**) V1_02_medium (**f**) V1_03_difficult

Figure 12. Comparison of trajectory error.

The absolute trajectory error APE is represented as a color. The trajectory calculated in this paper was relatively close to the true value trajectory in the dataset, but the cumulative error increased with the distance. From the trajectory calculation effect of each sub-sequence, the cumulative error was smaller in the sequence with good texture, good lighting, slow movement, and good lighting (the red trajectory is less). It showed that motion and lighting conditions have a great influence on the accuracy of the algorithm. APE was used to quantitatively analyze the pose error obtained by the algorithm in this paper, and to compare it with the trajectory error obtained by the VINS-Mono algorithm, as shown in Table 2.

Table 2. APE root mean square error(RMSE) in EuRoc datasets.

Sequence	Track Length	Test Conditions	RMSE	
			VINS-Mono	Ours
MH_01_easy	80.6 m	Situation A [1]	0.137 m	**0.122 m**
MH_02_easy	73.5 m	Situation A	0.143 m	**0.134 m**
MH_03_medium	130.9 m	Situation B [2]	2.263 m	**0.155 m**
MH_04_difficult	91.7 m	Situation C	0.362 m	**0.347 m**
MH_05_difficult	97.6 m	Situation C [3]	0.377 m	**0.302 m**
V1_01_easy	58.6 m	Situation D [4]	**0.080 m**	0.087 m
V1_02_medium	75.9 m	Situation B	0.201 m	**0.110 m**
V1_03_difficult	79.0 m	Situation C	0.201 m	**0.187 m**
V2_01_easy	36.5 m	Situation A	0.088 m	**0.086 m**
V2_02_medium	83.2 m	Situation B	0.158 m	**0.148 m**
V2_03_difficult	86.1 m	Situation C	0.307 m	**0.277 m**

[1] Test equipment moving slowly in good light. [2] Test equipment moving fast in good light. [3] Test equipment moving fast in poor light. [4] Test equipment moving slowly in poor light.

The visual IMU fusion positioning and mapping algorithm designed in this paper were deployed on the Manifold-2C computing platform, which was configured with Intel Core i7-8550U; 8 GB 64 bit DDR4 2400 MHz RAM; 256 GB SSD, using Intel's RealSense D435i camera as the sensor. The D435i output maximum resolution was 1920 × 1080 p, 30 fps. The D435i integrated an IMU model BMI055, and the output frequency of the IMU was 200 Hz. The camera and IMU were calibrated by Kalibr and imu_utils tools, and the relevant parameters of the calibration are shown in Table 3.

Table 3. Parameters of the camera and IMU.

Item	Parameters
Camera internal parameters	$\begin{bmatrix} 607.78 & 0 & 321.35 \\ 0 & 608.07 & 236.30 \\ 0 & 0 & 1 \end{bmatrix}$
Camera distortion parameters $\begin{bmatrix} k_1 & k_2 & p_1 & p_2 \end{bmatrix}$	$\begin{bmatrix} 0.0765 & 0.0185 & 0.00006387 & 0.00003974 \end{bmatrix}$
Accelerometer noise	0.0187 m/s^2
Accelerometer Random Walk	0.000596 m/s^2
Gyroscope noise	0.0018 rad/s
Angular Random Walk	0.000011 rad/s

Compared with the small scene in the indoor laboratory, the coal mine tunnel scene has poor lighting and sparse features, which can better test the robustness of the algorithm. In order to evaluate the performance of the fusion algorithm in special scenarios, tests were carried out in coal mine tunnels. The experimental environment was a simulated tunnel of a coal mine in the school, and the scene is shown in Figure 13.

Because there was no motion capture device to obtain the trajectory of motion in the coal tunnel, we used the laser SLAM results as the true value. We used the EVO tool to plot the trajectories of the proposed method in this paper and VINS_Mono under two paths as shown in Figure 14. Where the dashed line represents the true value of the trajectory, the blue line represents the trajectory calculated by VINS_Mono, and the improved method in this paper is represented by the green line. Table 4 shows the APE between the trajectories and the true values.

Figure 13. Scenes from coal mine tunnel.

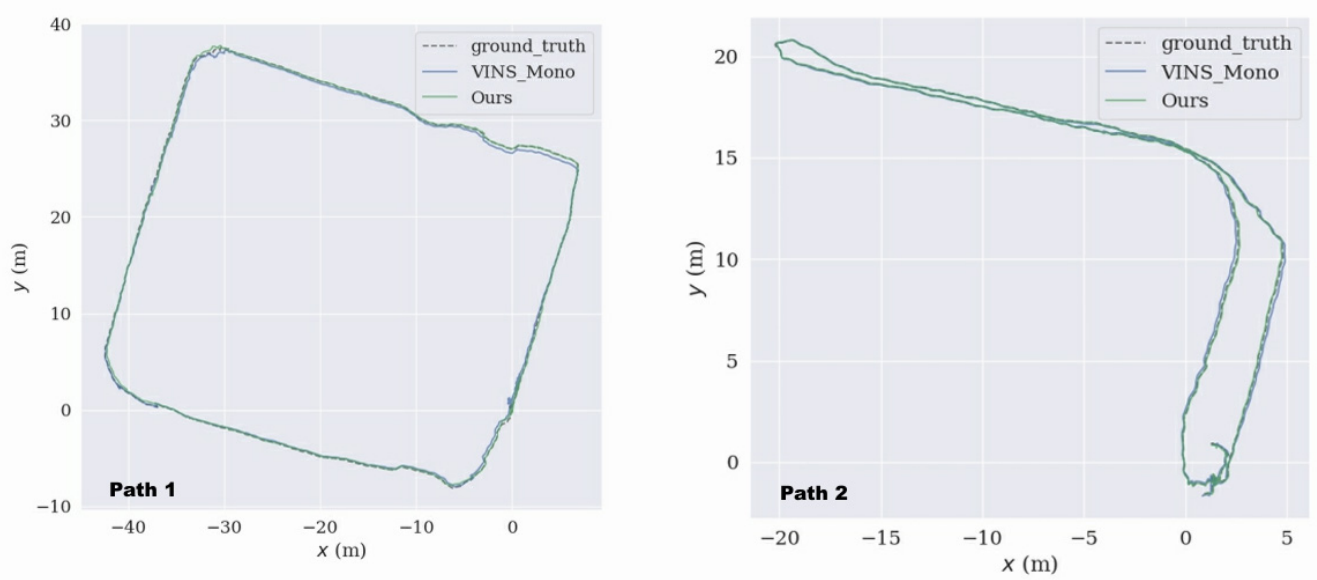


Figure 14. Trajectory error between the two algorithms and the true value under two paths.

Table 4. Error statistics between the two algorithms and the true value under two paths.

APE	Path1		Path2	
	VINS_Mono	Ours	VINS_Mono	Ours
Max	0.879 m	**0.660 m**	0.219 m	**0.207 m**
Mean	0.330 m	**0.192 m**	0.120 m	**0.046 m**
Min	0.028 m	**0.005 m**	0.025 m	**0.004 m**
RMSE	0.370 m	**0.231 m**	0.129 m	**0.052 m**
Std	0.167 m	**0.130 m**	0.047 m	**0.026 m**

The above table shows the maximum deviation, average deviation, minimum deviation, RMSE, and standard deviation of the trajectories estimated on the two paths,

respectively, for each of the two different algorithms for the relative translations. The best results are shown in bold. Numerically, the proposed method achieved the RMSE of the estimated tracking trajectory as low as 0.231 m and 0.052 m for both paths, respectively, which outperforms the comparative algorithms of 0.37 m and 0.129 m. It indicates that the proposed method in this paper achieves a higher localization accuracy under both coal mine tunnel paths.

4. Conclusions

In this paper, we propose a localization and mapping method for coal mine tunnel localization incorporating visual features and IMU, which uses data from IMU to assist monocular cameras for motion estimation. We discarded optical flow tracking and used ORB features and line features for feature matching to reduce feature loss due to fast motion. IMU provided motion compensation to reduce cumulative error. The robustness of the visual-inertial odometer in environments with poor lighting conditions was increased by closely coupling the IMU with the image features. Finally, by introducing closed-loop detection, visual information was fully utilized to obtain more accurate sensor motion trajectories. Experiments were conducted on the EuRoc dataset to compare the trajectories of the algorithm proposed in this paper with the VINS-Mono algorithm. The actual environment is verified in a simulated coal mine tunnel. The results showed that the trajectories obtained by the algorithm proposed in this paper were more accurate and more robust in the scenarios of a coal mine tunnel.

Author Contributions: Conceptualization, D.Z.; Methodology, D.Z. and K.J.; Data curation, K.J. and D.W.; Formal analysis, S.L.; Investigation, D.Z.; Project administration, D.Z. and S.L.; Software, K.J. and D.W.; Supervision, D.Z.; Writing—original draft, K.J. and S.L.; Writing—review and editing, S.L. and K.J. All authors have read and agreed to the published version of the manuscript.

Funding: This study was supported by the National Natural Science Foundation of China (51774235) and the Shaanxi Provincial Key R&D General Industrial Project (2021GY-338).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

References

1. Cadena, C.; Carlone, L.; Carrillo, H.; Latif, Y.; Scaramuzza, D.; Neira, J.; Reid, I.; Leonard, J.J. Past, Present, and Future of Simultaneous Localization and Mapping: Toward the Robust-Perception Age. *IEEE Trans. Robot.* **2016**, *32*, 1309–1332. [CrossRef]
2. 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.
3. Ding, W.; Hou, S.; Gao, H.; Wan, G.; Song, S. LiDAR Inertial Odometry Aided Robust LiDAR Localization System in Changing City Scenes. In Proceedings of the 2020 IEEE International Conference on Robotics and Automation (ICRA), Paris, France, 31 May–31 August 2020; pp. 4322–4328.
4. Tarokh, M.; Merloti, P.; Duddy, J.; Lee, M. Vision Based Robotic Person Following under Lighting Variations. In Proceedings of the 2008 3rd International Conference on Sensing Technology, Taipei, Taiwan, 30 November–3 December 2008; pp. 147–152.
5. Lowe, D.G. Distinctive Image Features from Scale-Invariant Keypoints. *Int. J. Comput. Vis.* **2004**, *60*, 91–110. [CrossRef]
6. Bay, H.; Tuytelaars, T.; Van Gool, L. SURF: Speeded Up Robust Features. In *Computer Vision—ECCV 2006*; Leonardis, A., Bischof, H., Pinz, A., Eds.; Lecture Notes in Computer Science; Springer Berlin Heidelberg: Berlin/Heidelberg, Germany, 2006; Volume 3951, pp. 404–417. ISBN 978-3-540-33832-1.
7. Harris, C.; Stephens, M. A Combined Corner and Edge Detector. In Proceedings of the Fourth Alvey Vision Conference, Manchester, UK, 31 August–2 September 1988; pp. 147–152.
8. Rublee, E.; Rabaud, V.; Konolige, K.; Bradski, G. ORB: An Efficient Alternative to SIFT or SURF. In Proceedings of the 2011 International Conference on Computer Vision, Barcelona, Spain, 6–13 November 2011; pp. 2564–2571.

9. Mur-Artal, R.; Tardos, J.D. ORB-SLAM2: An Open-Source SLAM System for Monocular, Stereo, and RGB-D Cameras. *IEEE Trans. Robot.* **2017**, *33*, 1255–1262. [CrossRef]
10. Engel, J.; Koltun, V.; Cremers, D. Direct Sparse Odometry. *IEEE Trans. Pattern Anal. Mach. Intell.* **2018**, *40*, 611–625. [CrossRef] [PubMed]
11. Li, J.; Yang, B.; Huang, K.; Zhang, G.; Bao, H. Robust and Efficient Visual-Inertial Odometry with Multi-Plane Priors. In *Pattern Recognition and Computer Vision*; Lin, Z., Wang, L., Yang, J., Shi, G., Tan, T., Zheng, N., Chen, X., Zhang, Y., Eds.; Lecture Notes in Computer Science; Springer International Publishing: Cham, Switzerland, 2019; Volume 11859, pp. 283–295. ISBN 978-3-030-31725-6.
12. Xie, X.; Zhang, X.; Fu, J.; Jiang, D.; Yu, C.; Jin, M. Location Recommendation of Digital Signage Based on Multi-Source Information Fusion. *Sustainability* **2018**, *10*, 2357. [CrossRef]
13. Pumarola, A.; Vakhitov, A.; Agudo, A.; Sanfeliu, A.; Moreno-Noguer, F. PL-SLAM: Real-Time Monocular Visual SLAM with Points and Lines. In Proceedings of the 2017 IEEE International Conference on Robotics and Automation (ICRA), Singapore, 29 May–3 June 2017; pp. 4503–4508.
14. Campos, C.; Elvira, R.; Rodriguez, J.J.G.; Montiel, J.M.; Tardos, J.D. ORB-SLAM3: An Accurate Open-Source Library for Visual, Visual–Inertial, and Multimap SLAM. *IEEE Trans. Robot.* **2021**, *37*, 1874–1890. [CrossRef]
15. Von Stumberg, L.; Usenko, V.; Cremers, D. Direct Sparse Visual-Inertial Odometry Using Dynamic Marginalization. In Proceedings of the 2018 IEEE International Conference on Robotics and Automation (ICRA), Brisbane, QLD, Australia, 21–25 May 2018; pp. 2510–2517.
16. Mourikis, A.I.; Roumeliotis, S.I. A Multi-State Constraint Kalman Filter for Vision-Aided Inertial Navigation. In Proceedings of the Proceedings 2007 IEEE International Conference on Robotics and Automation, Rome, Italy, 10–14 April 2007; pp. 3565–3572.
17. Leutenegger, S.; Lynen, S.; Bosse, M.; Siegwart, R.; Furgale, P. Keyframe-Based Visual–Inertial Odometry Using Nonlinear Optimization. *Int. J. Robot. Res.* **2015**, *34*, 314–334. [CrossRef]
18. 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]
19. Burri, M.; Nikolic, J.; Gohl, P.; Schneider, T.; Rehder, J.; Omari, S.; Achtelik, M.W.; Siegwart, R. The EuRoC Micro Aerial Vehicle Datasets. *Int. J. Robot. Res.* **2016**, *35*, 1157–1163. [CrossRef]
20. Allak, E.; Jung, R.; Weiss, S. Covariance Pre-Integration for Delayed Measurements in Multi-Sensor Fusion. In Proceedings of the 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Macau, China, 3–8 November 2019; pp. 6642–6649.
21. Deschaud, J.-E. IMLS-SLAM: Scan-to-Model Matching Based on 3D Data. In Proceedings of the 2018 IEEE International Conference on Robotics and Automation (ICRA), Brisbane, QLD, Australia, 21–25 May 2018; pp. 2480–2485.
22. Hutchison, D.; Kanade, T.; Kittler, J.; Kleinberg, J.M.; Mattern, F.; Mitchell, J.C.; Naor, M.; Nierstrasz, O.; Pandu Rangan, C.; Steffen, B.; et al. Adaptive and Generic Corner Detection Based on the Accelerated Segment Test. In *Computer Vision—ECCV 2010*; Daniilidis, K., Maragos, P., Paragios, N., Eds.; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2010; Volume 6312, pp. 183–196. ISBN 978-3-642-15551-2.
23. Galvez-López, D.; Tardos, J.D. Bags of Binary Words for Fast Place Recognition in Image Sequences. *IEEE Trans. Robot.* **2012**, *28*, 1188–1197. [CrossRef]
24. Hutchison, D.; Kanade, T.; Kittler, J.; Kleinberg, J.M.; Mattern, F.; Mitchell, J.C.; Naor, M.; Nierstrasz, O.; Pandu Rangan, C.; Steffen, B.; et al. BRIEF: Binary Robust Independent Elementary Features. In *Computer Vision—ECCV 2010*; Daniilidis, K., Maragos, P., Paragios, N., Eds.; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2010; Volume 6314, pp. 778–792. ISBN 978-3-642-15560-4.
25. von Gioi, R.G.; Jakubowicz, J.; Morel, J.-M.; Randall, G. LSD: A Fast Line Segment Detector with a False Detection Control. *IEEE Trans. Pattern Anal. Mach. Intell.* **2010**, *32*, 722–732. [CrossRef] [PubMed]
26. Zhang, L.; Koch, R. An Efficient and Robust Line Segment Matching Approach Based on LBD Descriptor and Pairwise Geometric Consistency. *J. Vis. Commun. Image Represent.* **2013**, *24*, 794–805. [CrossRef]
27. Brickwedde, F.; Abraham, S.; Mester, R. Mono-SF: Multi-View Geometry Meets Single-View Depth for Monocular Scene Flow Estimation of Dynamic Traffic Scenes. In Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Korea, 17–19 October 2019; pp. 2780–2790.
28. Vijayanarasimhan, S.; Ricco, S.; Schmid, C.; Sukthankar, R.; Fragkiadaki, K. SfM-Net: Learning of Structure and Motion from Video. *arXiv* **2017**, arXiv:1704.07804.
29. Coulin, J.; Guillemard, R.; Gay-Bellile, V.; Joly, C.; de La Fortelle, A. Tightly-Coupled Magneto-Visual-Inertial Fusion for Long Term Localization in Indoor Environment. *IEEE Robot. Autom. Lett.* **2022**, *7*, 952–959. [CrossRef]
30. Mistry, M.; Letsios, D.; Krennrich, G.; Lee, R.M.; Misener, R. Mixed-Integer Convex Nonlinear Optimization with Gradient-Boosted Trees Embedded. *INFORMS J. Comput.* **2021**, *33*, 1103–1119. [CrossRef]
31. Braverman, V.; Drineas, P.; Musco, C.; Musco, C.; Upadhyay, J.; Woodruff, D.P.; Zhou, S. Near Optimal Linear Algebra in the Online and Sliding Window Models. In Proceedings of the 2020 IEEE 61st Annual Symposium on Foundations of Computer Science (FOCS), Durham, NC, USA, 16–19 November 2020; pp. 517–528.

32. MacTavish, K.; Paton, M.; Barfoot, T.D. Visual Triage: A Bag-of-Words Experience Selector for Long-Term Visual Route Following. In Proceedings of the 2017 IEEE International Conference on Robotics and Automation (ICRA), Singapore, 29 May–3 June 2017; pp. 2065–2072.
33. Forster, C.; Carlone, L.; Dellaert, F.; Scaramuzza, D. IMU Preintegration on Manifold for Efficient Visual-Inertial Maximum-a-Posteriori Estimation. 10 May 2015; p.10. Available online: http://hdl.handle.net/1853/55417 (accessed on 9 August 2022).