A Robust Parallel Initialization Method for Monocular Visual-Inertial SLAM

Min Zhong, Yiqing Yao, Xiaosu Xu * and Hongyu Wei

Key Laboratory of Micro-Inertial Instruments and Advanced Navigation Technology, Ministry of Education, School of Instrument Science and Engineering, Southeast University, Nanjing 210096, China
* Correspondence: xxs@seu.edu.cn

Abstract: In order to improve the initialization robustness of visual inertial SLAM, the complementarity of the optical flow method and the feature-based method can be used in vision data processing. The parallel initialization method is proposed, where the optical flow inertial initialization and the monocular feature-based initialization are carried out at the same time. After the initializations, the state estimation results are jointly optimized by bundle adjustment. The proposed method retains more mapping information, and correspondingly is more adaptable to the initialization scene. It is found that the initialization map constructed by the proposed method features a comparable accuracy to the one constructed by ORB-SLAM3 in monocular inertial mode. Since the online extrinsic parameter estimation can be realized by the proposed method, it is considered better than ORB-SLAM3 in the aspect of portability. By the experiments performed on the benchmark dataset EuRoC, the effectiveness and robustness of the proposed method are validated.

Keywords: SLAM; initialization; visual-inertial; monocular

1. Introduction

In the research field of multi-sensor fusion simultaneous localization and mapping (SLAM), the fusion of vision and inertia has always been a hot research topic in recent years [1–8]. The rich environmental information captured by image can be used for building maps and estimating the up-to-scale trajectory of the camera, which is attached to the mobile platform. The measurements of gyroscope and accelerometer contained in the inertial measurement unit (IMU) sensor can be integrated to estimate accurate short-term rigid body motion. In addition, both the camera and the IMU have matured miniaturized products. These properties make the combination of the monocular vision camera and the IMU economical, effective, and low energy for localization and mapping in small mobile robot or other platforms with limited load.

The state-of-the-art visual-inertial SLAM methods [1–8] presented in the literature widely use the IMU preintegration [9] and the keyframe-based nonlinear optimization methods [6]. However, the performance of these techniques heavily depends on prior precise extrinsic calibration of the six-degree-of-freedom (6-DOF) transformation between the camera and the IMU. As the coordinate conversion parameters between the camera and the IMU, the extrinsic parameters play important roles in accurate data fusion between the camera and the IMU. Incorrect extrinsic calibration leads to system deviation and degrades the overall navigation performance. For the acquisition of extrinsic parameters, most of the visual-inertial SLAM methods [1–7] use Kalib to calibrate the extrinsic parameters, which a state-of-the-art marker-based off-line calibration method [10]. Thereafter, the calibrated extrinsic parameters by Kalib are considered accurate and constant during the running of the whole program. However, this process is complex and time consuming. Additionally, it is usually required to repeat this process when the sensors are repositioned. Therefore, in order to improve the portability and operability of the program, an automatic extrinsic parameter estimation is necessary. At present, the automatic extrinsic parameter estimation methods are not mature enough.
calibration [11] has been implemented in a monocular visual-inertial system constructed by one camera and a low-cost IMU (VINS-Mono) [8]. Weiho Huang et al. [12] implemented self-calibration of the camera and the IMU extrinsic parameters in the monocular ORB feature-based SLAM system (ORB-SLAM) framework [1,2]; ORB are binary features invariant to rotation and scale (in a certain range), resulting in a very fast recognizer with good invariance to viewpoint, although this code is not open source.

For visual measurement processing, it can be divided into the optical flow methods [13] and the feature-based methods [14–16] according to the matching strategy.

The feature-based methods represented by ORB-SLAM series [1–4] extract the ORB feature points [14] from the image, then calculate the descriptor of the corresponding feature points, and finally obtain the feature point pairs in two frames through the descriptor comparison. This kind of method features high positioning accuracy, but it is time-consuming and easy to lose tracking in a weak texture environment or due to blurred image in fast motion [17].

The same as feature-based methods, the optical flow methods represented by VINS-Mono also extract feature points from the image. However, the latter ones use image pixel values to match feature points, and eliminate the descriptor calculation of feature points, which is the most time-consuming process in feature-based methods. This characteristic makes the optical flow methods suitable for real-time and resource-constrained occasions, while the accuracy is also guaranteed to a certain extent. Moreover, the robustness to pixel noise of the optical flow methods is better than feature-based methods.

It must be noted that, due to the assumption of photometric invariance [14], the optical flow methods are easily affected by external illumination, which means the feature-based methods feature a better robustness to external illumination than optical flow methods. Hence, a mixed method can be built based on the optical flow and feature-based methods, in order to achieve their robustness complementarity. The optical flow methods complement the feature point method when the image descriptor cannot be accurately calculated due to motion blur. The feature point methods provide better inter-frame pose results to suppress the motion drift caused by the wrong feature matching of the optical flow method.

Although the robustness complementarity has been achieved in the mixed method, the risk of tracking loss in weak illumination and low texture environment cannot be eliminated. Thus, the participation of IMU is very necessary.

The involvement of IMU data needs to go through an initialization process [18,19], which mainly completes the work of finding the gravity direction and correcting the bias of gyroscopes and accelerometers [20]. The initialization of IMU data has been implemented by many algorithms [1–9]. Nowadays, most of the state-of-the-art methods use the visual initialization information as a reference to correct the gyroscope and accelerometer bias of IMU [4–8] at the start. Taking VINS-Mono as an example [19], the initialization of the IMU is realized by aligning the IMU preintegration results with the vision-only structure from motion (SfM) results, which can obtain a rough estimation of IMU state vector. Then, the related parameters of the IMU are continuously optimized in the subsequent tracking process. This leads to the fact that the accuracy of VINS-Mono may not be ideal when the map is just initialized. For the monocular mode of ORB-SLAM2 [21], only two keyframes with sufficient matching feature points are needed for initialization, and the combination with IMU is not considered, so the initialization process is faster than that of VINS-Mono, but the map is constructed up-to-scale. Although the open-source code of ORB-SLAM3 [22] adds the mode of monocular camera with IMU, it also includes multi-map management and feature-based relocation and loop closing. The implementation of these functions relies on the continuous increase of feature saving and management with the expansion of the map range, which consumes a large amount of computer memory, making ORB-SLAM3 unable to run in real time on resource-constrained platforms. In addition, the self-calibration of extrinsic parameters is not considered in ORB-SLAM3.

Based on the above analysis, this paper proposes a robust parallel initialization method for monocular visual inertial SLAM, which integrates the characteristics of VINS-Mono
and monocular mode of ORB-SLAM2 in the initialization phase. Based on the ORB-SLAM2 initialization algorithm framework and the VINS-Mono initialization algorithm framework, the research work of this paper first executes the monocular initialization thread in ORB-SLAM2 and the monocular inertial initialization process in VINS-Mono in parallel, and then uses the Umeyama algorithm [23] and bundle adjustment (BA) [24] optimization to integrate their initialization data. The result of this method is more robust, and with higher accuracy than the original VINS-Mono initialization. Compared with the original monocular initialization trajectory of ORB-SLAM2, the trajectory obtained by the initialization in this paper is in metric unit instead of up-to-scale, which better describes the real external environment. It also retains the online extrinsic parameter calibration function in VINS-Mono initialization, which improves the portability of the initialization algorithm. In the subsequent experimental comparison with the state-of-the-art ORB-SLAM3 initialization results, it can be found that the accuracy of the map obtained by the two initializations is comparable. The contributions can be summarized as follows:

- The design of two-thread parallel initialization makes SLAM system have better portability and initialization robustness.
- The designed initialization mode improves the initialization accuracy of the original VINS-Mono, which can provide a more accurate initial value of state estimation to the visual inertial SLAM system.
- The initialization state estimated by the proposed method makes the trajectory accurate like the one estimated by ORB-SLAM3 in monocular-inertial mode.

The remainder of this article is arranged as follows: In Section 2, the measurement preprocessing of the IMU and the camera information are described. In Section 3, the detail of the initialization process in the proposed method is introduced. Section 4 presents the experimental results of the improved initialization of monocular visual-inertial SLAM. Finally, the conclusions of this study are presented in Section 5.

2. Measurement Preprocessing

To implement the initialization method proposed in this paper, monocular camera and IMU measurements need to be preprocessed. For monocular visual measurements, we use the combination of the optical flow method and the feature-based method to track features between consecutive frames and detect new features in the latest frame. For IMU measurements, the outputs of IMU are pre-integrated between consecutive frames.

2.1. Vision Preprocessing

The optical flow tracking algorithm and ORB feature-based method are carried out in parallel for each new image. In both of them, keyframes are selected and saved according to the tracking effect. The optical flow method judges keyframes according to the average disparity between the current frame and the previous keyframe. The feature-based method is judged by the number of tracked feature points.

2.1.1. Optical Flow Tracking Algorithm

The main function of the optical flow tracking algorithm in this paper is to match the feature points between successive image frames based on the optical flow method for the feature points extracted from image data, and to get the feature matching relationship between image frames.

Specifically, for each new frame, new corner feature points were detected after existing feature points were tracked by Kanade–Lucas–Tomasi (KLT) sparse optical flow algorithm [25], and this operation is to ensure the number of feature points (100–300) detected in each frame are sufficient. Then, outlier rejection is performed using RANSAC [26] with a fundamental matrix model.

KLT corner tracking algorithm is also known as LK tracking algorithm. It is a classic corner tracking algorithm. The algorithm assumes that the target is in the video stream, only a small consistent displacement is generated, and the gray level of the target does
not change much. The detailed description of the LK tracking algorithm is attached in Appendix A.

2.1.2. ORB Feature-Based Method

The feature-based method includes detecting and matching steps of feature points. The form of feature points adopts ORB feature points [14], which include FAST corner extraction and BRIEF descriptor calculation.

The main role of the method based on ORB feature in this paper is to obtain the motion relationship between two frames as well. The method is to calculate the matching relationship of ORB feature points extracted between two frames, and then to establish the initialized map according to the motion relationship and the position information of feature points obtained by the principle of epipolar geometry. The matching relationship is achieved by calculating the Hamming distance of the ORB descriptor of the relevant feature points in the image.

For instance, when the similarity between the descriptor of feature point A in image frame 1 and that of feature point B in image frame 2 is greater than 90%, A and B are considered to be the same feature point, that is, the two points are successfully matched. Since the motion relationship between image frame 1 and image frame 2 is unknown during the initialization process, we usually match all feature points within a certain range of pixel coordinates of feature point A of image 2 to obtain feature point B that matches feature point A in image frame 1.

2.2. IMU Preprocessing

IMU senses the information of acceleration and angular velocity of motion and provides relative positioning information. The combination of IMU and monocular camera can provide an absolute scale reference of the real environment for the SLAM system. The derivation process of preintegration is consistent with subsection B of section IV in VINS-Mono [27].

2.2.1. Model of Output

An IMU measures the rotation rate and the acceleration of the sensors with respect to an inertial frame. The measurements, namely $\hat{a}_t$ and $\hat{\omega}_t$, are affected by additive white noise $n$ and a slowly varying sensor bias $b$. Considering accelerometer bias $b_a$, gyroscope bias $b_\omega$, and additive noise, the raw measurements of accelerometer and gyroscope, $\hat{a}$ and $\hat{\omega}$, are separately modeled as follows:

$$\begin{align*}
\hat{a}_t &= a_t + b_a + R_t^w g^w + n_a, \\
\hat{\omega}_t &= \omega_t + b_\omega + n_\omega
\end{align*}$$

The first items on the right side of (1) are the true value of accelerometer and gyroscope, which are all three-axis measurement information, as shown in (3), the second term is the bias, and the last term is the measurement noise. When the accelerometer is stationary, it is affected by the gravity acceleration value in the world coordinate system $g^w$.

The additive noise in acceleration and gyroscope measurements are assumed as Gaussian white noise, $n_a \sim \mathcal{N}(0, \sigma^2_a)$, $n_\omega \sim \mathcal{N}(0, \sigma^2_\omega)$. Acceleration bias and gyroscope bias are modeled as random walk, $b_a = n_b_a$, $b_\omega = n_b_\omega$, whose derivatives are Gaussian white noise, $n_{b_a} \sim \mathcal{N}(0, \sigma^2_{b_a})$, $n_{b_\omega} \sim \mathcal{N}(0, \sigma^2_{b_\omega})$. The $t$ subscript in the corresponding parameter denotes the value of the parameter at time $t$. $R_t^w$ represents the rotation matrix from the world coordinate system to the current inertial coordinate system.
2.2.2. IMU Preintegration

For two consecutive frames $b_k$ and $b_{k+1}$, there exists several inertial measurements in time interval $[t_k, t_{k+1}]$. Given the bias estimation, we integrate them in local frame $b_k$ as:

\[
\begin{align*}
\dot{a}_{b_k}^k &= \int_{t_k}^{t_{k+1}} R_t^b (\ddot{a}_t - b_{a_t}) dt^2 \\
\dot{\beta}_{b_k}^k &= \int_{t_k}^{t_{k+1}} R_t^b (\ddot{a}_t - b_{a_t}) dt \\
\dot{\gamma}_{b_k}^k &= \int_{t_k}^{t_{k+1}} \frac{1}{2} \Omega(\ddot{a}_t - b_{a_t}) \gamma_{b_t}^k dt
\end{align*}
\]

(2)

where $\gamma_{b_k}^k$ is the quaternion representation of angular velocity increment, among

\[
\Omega(\omega) = \begin{bmatrix}
-\omega_y & \omega_z & 0 \\
-\omega_z & -\omega_x & \omega_y \\
\omega_z & -\omega_y & -\omega_x
\end{bmatrix}
\]

\[
[\omega]_\times = \begin{bmatrix}
0 & -\omega_z & \omega_y \\
\omega_z & 0 & -\omega_x \\
-\omega_y & \omega_x & 0
\end{bmatrix}
\]

According to the output of accelerometer and gyro in Equation (1), $a_{b_{k+1}}$ and $b_{b_{k+1}}$, and $\gamma_{b_{k+1}}$ are the velocity increment pre-integrated measurements, the position increment pre-integrated measurements, and the rotation amount pre-integrated measurements of the motion between consecutive frames $b_k$ and $b_{k+1}$, respectively. It can be seen that the preintegration term (2) can be solely with IMU measurements by taking $b_k$ as the reference frame given bias. $R_t^b$ is the rotation matrix from the inertial frame at time $t$ to the inertial frame of $b_k$.

2.2.3. Bias Correction

Equation (2) is derived basing on the assumption that the bias of gyro and additive table is constant in the integration interval, while in fact the bias changes slowly in the output process of IMU. When the bias changes, if the measurement is still calculated according to Equation (2), the pre-integral measurements need to be recomputed, which is computationally expensive. To solve this problem, if the estimation of bias changes minorly, we adjust $a_{b_{k+1}}$, $b_{b_{k+1}}$, and $\gamma_{b_{k+1}}$ by their first-order approximations with respect to the bias as:

\[
\begin{align*}
\dot{a}_{b_k}^k &\approx \dot{a}_{b_k}^k + \int_{t_k}^{t_{k+1}} \Omega_{a_k} \delta b_{a_k} dt + \int_{t_k}^{t_{k+1}} \Omega_{a_k} \delta b_{a_t} dt \\
\dot{\beta}_{b_k}^k &\approx \dot{\beta}_{b_k}^k + \int_{t_k}^{t_{k+1}} \Omega_{a_k} \delta b_{a_k} dt + \int_{t_k}^{t_{k+1}} \Omega_{a_k} \delta b_{a_t} dt \\
\dot{\gamma}_{b_k}^k &\approx \dot{\gamma}_{b_k}^k + \frac{1}{2} \Omega_{a_k} \delta b_{a_t}
\end{align*}
\]

(4)

where $\Omega_{a_k}$ denotes the Jacobian matrix of $y$ versus $x$; refer to VINS-Mono for specific definitions. At the beginning, $a_{b_k}$ and $b_{b_k}$, and $\gamma_{b_k}$ are 0, and $\gamma_{b_k}$ is identity quaternion. Moreover, $\otimes$ is a quaternion multiplication operator.

3. The Initialization Method Integrates Feature-Based Pose Information, Optical Flow Pose Information and Inertial Information

If we process the results of the optical flow method, the feature-based method, and IMU preintegration at the same time, the amount of data to be optimized is considerable, which requires high computer performance; therefore, we use parallel threads to initialize and a two-step optimization. In order to make full use of the existing open-source programs and reduce the amount of calculation of the programs at the same time, we first initialize the optical flow data and feature-based data separately, and then fuse the two initialization results to complete the combination of visual information as well as the inertial information.

The specific implementation process is shown in Figure 1. Firstly, the initialization threads based on feature points and optical flow method are carried out in parallel. After the completion of the initialization of the two threads, the coarse alignment of the initial
map is realized by using the Umeyama algorithm. Then, the local BA fusion of visual items is performed. Finally, the visual and inertial items in the two threads are fused to obtain the optimized initialization results.

Figure 1. The framework of the robust parallel initialization methods.

3.1. Initialization Framework for Monocular VIO in the Optical Flow Method

The optical flow monocular inertial initialization method adopted by VINS-Mono is advanced and widely used. It also includes the extrinsic parameters estimation [11] between the camera and the IMU, and the alignment of the world coordinate with the gravity direction. The procedure of initialization framework for VINS is shown in Figure 2.

Figure 2. The initialization framework of VINS-Mono, including extrinsic calibration.

Visual tracking based on optical flow method requires continuous frames and small movement distance. However, in order to obtain the initial map points, the matching feature points need to be triangulated, and the premise of triangulation is that there is a large enough displacement between the two image frames where the feature points are located. Therefore, the initialization based on the optical flow method tends to establish a sliding window of size 10 as a data buffer to save the image frame, and the saving unit is one image frame. The initialized operation will find two frames with appropriate disparity in the sliding window for triangulation and propagate the triangulation results to other frames in the sliding window based on the results of optical flow tracking. Note that only frames that are successfully tracked and have a certain difference from the previous frames are used for triangulation.
frame will be added to the sliding window. In VINS-Mono, section VI, Part D, has specific operations on marginalization of state items in the sliding window.

3.1.1. Vision-Only SfM in Sliding Window

As can be seen in Figure 2, the initialization method based on the optical flow method starts on the premise that the sliding window, which serves as the image frame data buffer, has been filled with the image frames output by the camera. When the sliding window is full of frames, we check feature correspondences between the latest frame and all previous frames. If we can find stable feature tracking (more than 30 tracked features) and sufficient parallax (more than 20 pixels) between the latest frame and any other frames in the sliding window, we recover the relative rotation and up-to-scale translation between these two frames using the five-point algorithm [28]. Then, the feature points observed in the two frames are triangulated. Based on these triangulated features, a PnP method [29] is performed to estimate poses of all other frames in the window. Finally, a global full BA is carried on to minimize the total reprojection error of all feature observations. In this case, the first image frame in the sliding window is the reference coordinate system.

3.1.2. Visual-Inertial Alignment

In VINS-Mono, to scale the visual map to metric scale, the IMU preintegration is aligned with the SfM pose estimation results from the previous step, as show in Figure 3 [8]. The basic idea is to match the up-to-scale visual structure with IMU preintegration.

According to the spatial transformation relationship, we can easily know the coordinate transformation relationship between the camera frame and IMU frame as follows:

\[
[q_{b_i}^b \otimes q_{c_i}^c = q_{c_i}^c \otimes q_{b_{i+1}}^b] q_{c_i}^c = Q_{k+1}^i q_{c_i}^c = 0,
\]

(5)

where

\[
Q_1(q) = \begin{bmatrix} q_w I_3 + [q_{xyz} \times] & q_{xyz} \\ -q_{xyz} & q_w \end{bmatrix},
\]

\[
Q_2(q) = \begin{bmatrix} q_w I_3 - [q_{xyz} \times] & q_{xyz} \\ -q_{xyz} & q_w \end{bmatrix},
\]

(6)

are matrix representation for left and right quaternion multiplication, \([q_{xyz} \times]\) is the skew-symmetric matrix from the first three elements \(q_{xyz}\) of a quaternion, \(q_w\) is the real part of the quaternion. Moreover, \(q_{c_i}^c\) denotes the rotation quaternion from the \(y\) coordinate system to the \(x\) coordinate system, where \(c_i\) and \(b_i\) at positions of \(x\) and \(y\) represent the camera coordinate system and inertial coordinate system corresponding to the \(i\)-th frame image, respectively.
The relation of Equation (5) holds between all adjacent frames, and we can construct the overdetermined equation according to the equality relationship between multiple frames as:

\[
\begin{bmatrix}
  w^b_1 Q^0_1 \\
  w^b_2 Q^0_2 \\
  \vdots \\
  w^b_{N-1} Q^0_{N-1}
\end{bmatrix}
q^b_c = Q_N q^b_c = 0,
\]

(7)

where \( N \) is the index of the latest frame which keeps growing until the end of rotation calibration and \( w^b_k \) is weight derived from the robust norm for a better outlier processing. Then, use singular value decomposition (SVD) to obtain the rotation relationship between IMU and camera, \( q^b_c \). The extrinsic parameter results obtained by this method are not completely accurate because IMU error is not considered, and the value needs to be optimized by back-end optimization in the future. If \( q^b_c \) is known exactly, skip this step. See Formula (4)–(9) in [11] for the detailed calculation process.

- **Gyroscope Bias Calibration**

Combining the results of SfM, \( q^b_c \), and IMU preintegration in Equations (2) and (4), we can construct an equation between two consecutive frames, \( b_k \) and \( b_{k+1} \):

\[
q^c_{b_{k+1}} \otimes q^{0-1}_{b_k} = \gamma^{b_k}_{b_{k+1}},
\]

(8)

The cost equation is constructed accordingly, while the linear equation for IMU preintegration with respect to the gyro deviation is as follows to minimize the cost function:

\[
\min_{\delta b_k} \sum_{k \in B} \| q^{c}_{b_{k+1}} -1 \otimes q^{0}_{b_k} \otimes \gamma^{b_k}_{b_{k+1}} \|^2,
\]

(9)

where \( B \) indexes all frames in the window. In such a way, we get an initial calibration of the gyroscopic bias \( b_{\omega} \). Then, we repropagate all IMU preintegration terms \( \hat{\alpha}^{b_k}_{b_{k+1}}, \hat{\beta}^{b_k}_{b_{k+1}}, \) and \( \hat{\gamma}^{b_k}_{b_{k+1}} \) using the new gyroscopic bias.

- **Velocity, Gravity Vector, and Metric Scale Initialization**

We set the first camera frame \( (\cdot)^0 \) as the reference frame for SfM. All frame poses \( (p^0_i, q^0_i) \) and feature positions are represented with respect to \( (\cdot)^0 \). Moreover, \( (p^b_c, q^b_c) \) are extrinsic parameters between the camera and the IMU. After the gyroscopic bias is initialized, the gravity vector and metric scale in the world coordinate system can be initialized by establishing the relation equation between the IMU preintegration of two consecutive frames and the camera pose information obtained by SfM as:

\[
\begin{align*}
\alpha^{b_k}_{b_{k+1}} &= R^b_{c0} \left( s \left( p^0_{b_{k+1}} - p^0_{b_k} \right) + \frac{1}{2} g^0 \Delta t^2_k - R^b_{c0} v^b_{b_k} \Delta t^0_k \right), \\
\beta^{b_k}_{b_{k+1}} &= R^b_{c0} \left( R^0_{b_0} v^0_{b_{k+1}} + g^0 \Delta t_k - R^0_{b_k} v^0_{b_k} \right),
\end{align*}
\]

(10)

among

\[
\begin{align*}
q^0_{b_k} &= q^0_c \otimes (q^b_c)^{-1} \\
sp^0_{b_k} &= s^0 - R^b_{c0} p^b_c 
\end{align*}
\]

(11)

According to the above equation, and the velocity \( v^b_{b_k} \), gravity vector \( g^0 \) and scale \( s \) are taken as the state quantities to be estimated as:

\[
\chi I = \left[ v^b_{b_k}, v^b_{b_k}, g^0, s \right],
\]

(12)
Combine Equations (10) and (11) into the following linear measurement model:

$$\tilde{z}^b_{k+1} = \left[ a^b_{k+1} - p^b_k + R^b_0 R^{c_0}_{k+1} p^b_k \right] = H^b_{k+1} \chi I + n^b_{k+1},$$

(13)

where

$$H^b_{k+1} = \begin{bmatrix}
-I \Delta t_k & 0 & \frac{1}{2} R^b_0 \Delta t_k^2 & R^b_0 \left( \tilde{p}^b_{k+1} - \tilde{p}^b_k \right) \\
-I & R^b_0 R^{c_0}_{k+1} & R^b_0 & 0 \\
\end{bmatrix},$$

(14)

$R^b_0$, $R^{c_0}_{k+1}$, $\tilde{p}^b_{k+1}$, and $\tilde{p}^b_k$ are obtained from the up-to-scale monocular visual SfM, and $\Delta t_k$ is the time interval between two consecutive frames. By solving this linear least-square problem:

$$\min_{\chi} \sum_{k \in B} \| z^b_{k+1} - H^b_{k+1} \chi I \|^2,$$

(15)

the body frame velocities for every frame in the window, the gravity vector in the visual reference frame $(\cdot)^{c_0}$, as well as the scale parameter can be obtained.

- Gravity Refinement

Then, the gravity is perturbed with two variables on its tangent space, which preserves 2-DOF as in Equation (16). The aim is to optimize the direction of the gravity vector obtained in the initialization of the previous step:

$$g(\tilde{g} + \delta g), \delta g = \omega_1 b_1 + \omega_2 b_2,$$

(16)

where $g$ is the known magnitude of the gravity, and $\tilde{g}$ is the unit vector representing the direction of gravity. Moreover, $b_1$ and $b_2$ are two orthogonal bases spanning the tangent plane, as shown in Figure 4 [8], and $\omega_1$ and $\omega_2$ are 2-D perturbation toward $b_1$ and $b_2$, respectively. Lastly, $b_1$ and $b_2$ can be set arbitrarily as long as they are not parallel.

![Tangent space](image)

**Figure 4.** Illustration of 2-DOF perturbation of gravity. The magnitude of gravity is known.

Then, the $g$ in Equation (10) is replaced by $g(\tilde{g} + \delta g)$, and solve for 2-D $\delta g$ together with other state variables. This process iterates several times until $\tilde{g}$ converges.

- Completing Initialization

The rotation $q^c_0$ between the world frame and the camera frame $c_0$ can be obtained by rotating the refined gravity vector $\tilde{g}$ to the z-axis. Then, all variables from the reference frame $(\cdot)^c_0$ are rotated to the world frame $(\cdot)^w$. The body frame velocities will also be rotated to the world frame. Translational components from the visual SfM will be scaled to metric units. At this point, the initialization procedure is finished.

### 3.2. Initialization Framework for Monocular VO in ORB Feature-Based Method

After the detection and description of feature points in an image frame is finished, feature matching is carried out.

In this paper, the ORB initialization framework only carries out the pure vision initialization process as in ORB-SLAM2.
The monocular initialization process of ORB-SLAM2 can be divided into the following three stages.

3.2.1. Select two Frames as Initial Reference Frames

The first reference frame selected during initialization requires a sufficient number of feature points detected (>100); there is also a requirement on the number of feature point detection for the following image frames. If it does not meet the requirement on number of feature points in the two consecutive frames, the first reference frame should be selected again, and the above process should be repeated.

3.2.2. Calculation of Relative Pose Based on the Matching Features and Triangulation

When the number of feature points in two consecutive frames is sufficient, DBOW2 bag-of-word is used to accelerate the matching process of feature points. After word bag matching, multi-thread parallel processing is used to calculate the relative pose using basic matrix model and homography matrix model, respectively, and the pose transformation results of the two frames are obtained. The reprojection error of features is used to score the two pose results. The pose results obtained by the model with better score are selected to complete the triangulation of matching points. Then, the PnP algorithm is used to calculate the relative poses of the new image.

3.2.3. Global BA

Global BA is performed on the previously recovered feature points and poses to optimize the initialized map.

So far, we can obtain an initial up-to-scale map based on the ORB feature-based method that saves the relative poses and 3D coordinates of the corresponding feature points between image frames.

From the above description, we can see that the initialization process of the visual map based on ORB feature points is simple and fast, but the established map is not in metric units and does not align with the inertial coordinate system. On the premise that the two initialization processes are carried out synchronously, the map initialization based on ORB feature points should be completed first.

3.3. The Initialization Framework That Fuses the Initial Results of the Optical Flow Method and Feature Point Methods

The data fusion operation proposed in this paper will be performed after the initialization of both the optical flow method and the feature-based method. Since the initialization of the feature-based method only depends on the image of the monocular camera, the visual map established by feature-based method needs to be multiplied by a fixed scale to be in metric units. This map is referred to as the feature-based map in the following. So, the map established by optical flow method is referred to as the optical flow map.

The fusion process can be divided into three steps, as shown in Figure 5. First, the similarity matrix transformation from feature-based map is obtained by the Umeyama algorithm. The Umeyama algorithm is designed to compute the positional relationship between two sets of data. When aligning two trajectories \( x_i \) and \( y_i \), the goal is to compute a set of \( s, R, \) and \( t \) such that the objective function (17) is optimal:

\[
\frac{1}{n} \sum_{i=1}^{n} \| x_i - (sRy_i + t) \|_2^2, \quad (17)
\]

where \( s \) is the scaling factor, \( R \) is the rotation matrix, and \( t \) represents the translation.
Figure 5. The illustration of data fusion process between the feature-based map and the optical map.

Then, the transfer matrix between two co-view frames in the feature-based map is substituted into the corresponding relationship in optical flow method for optimization. Finally, the pose and rotation information of the state vector optimized in the previous step is fixed, and the remaining states in the state vector are further optimized.

3.3.1. Rough Map Alignment Using Umeyama Algorithm

In order to better perform the data fusion operation between the two maps, we use the Umeyama algorithm to solve the similarity transformation matrix, which is solved by the inputs of position information from the common key frame in the two maps. The basic idea of this step is to restore the feature-based map to metric unit and align with the world coordinate system in optical flow map through scaling, rotation, and translation operations.

Through the Umeyama algorithm, we can get the similarity transformation matrix

\[
\begin{bmatrix}
 s \mathbf{R}^\text{w}_k & t \\
 0 & 1
\end{bmatrix}
\]

and make the following equation between the two maps approximately true:

\[
\begin{align*}
\mathbf{P}^\text{w}_k & \approx s \mathbf{P}^\text{w}_k \mathbf{R}^\text{w}_k + t \\
\mathbf{R}_k & \approx \mathbf{R}_k \mathbf{R}^\text{w}_k
\end{align*}
\]  \hspace{1cm} (18)

The feature-based map can be approximate aligned with the optical flow map after the similarity transformation above, and the alignment process is shown in Figure 6.

Figure 6. Illustration of the alignment process between optical flow map and feature-based map by using Umeyama algorithm.

3.3.2. Local BA in Visual States

The feature-based map mainly saves the relative rotation and translation information between frames. Due to the higher accuracy of the feature matching algorithm used in the initialization of feature-based method, the optimization of the states in feature-based map
is ignored. The relevant information provided by the feature-based map is added to the optimization as measurement.

Thus, the state vector in the local BA is defined as:

$$
\mathbf{x}_L = \begin{bmatrix} y_0, y_1, \ldots, y_t, \hat{\mathbf{y}}, \ldots, \hat{\mathbf{y}}, \mathbf{w}_{\text{ins}}, \mathbf{w}_{\text{orb}} \end{bmatrix}^T,
$$

where $\hat{\mathbf{y}}$ is the position state of IMU states at the time the $k$th image is captured. It contains position and orientation of the IMU in the world frame. Moreover, $s$ is scale value in the similarity matrix which calculated by Umeyama algorithm and needs to be optimized in BA, $n$ is the total number of keyframes, and $m$ is the total number of features tracked by optical flow method in the sliding window. Lastly, $\gamma$ is the inverse distance of the $l$th features from its first observation.

The visual BA formulation can be optimized by minimizing the sum of the Mahalanobis norm of all measurement residuals to obtain a maximum posteriori estimation as:

$$
\min_{\mathbf{x}} \left\{ \sum_{(i,j) \in \Psi} \left( \left| \mathbf{r}_\psi \left( \mathbf{q}_{ij}^C, \mathbf{x}_L \right) \right| + \left| \mathbf{r}_\psi \left( \mathbf{P}_{ci}^{w_{\text{orb}}}, \mathbf{P}_{cj}^{w_{\text{orb}}} \right), \mathbf{X}_L \right| \right)^2 \right\} + \sum_{(i,j) \in \Omega} \left( \left| \mathbf{r}_\Omega \left( \mathbf{z}_{ij}^C, \mathbf{x}_L \right) \right| \right)^2 \right\},
$$

where the Huber norm is defined as:

$$
\rho(e) = \begin{cases} 
e & 1 \le e \le 2 \sqrt{2} - 1 \\ 2 \sqrt{2} - 1 & e > 1 \end{cases}
$$

where $\mathbf{r}_\psi \left( \mathbf{P}_{ci}^{w_{\text{orb}}}, \mathbf{P}_{cj}^{w_{\text{orb}}} \right), \mathbf{X}_L$, $\mathbf{r}_\psi \left( \mathbf{q}_{ij}^C, \mathbf{x}_L \right)$, and $\mathbf{r}_\Omega \left( \mathbf{z}_{ij}^C, \mathbf{x}_L \right)$ are residuals for scale estimation in feature-based map, rotation estimation between two common-view frames in feature-based map, and visual measurements in optical flow map, respectively. The detailed definition of residual term will be presented next.

- **Residuals for Rotation Estimation between Two Common-view Frames in feature-based Map**

The position parameters of the image frames in the optical flow map are mainly optimized based on the relative rotation between frames solved by the feature-based map. According to the inter-frame conversion matrix of the feature-based map, combined with Equation (18), the constrained equation holds as follow:

$$
\mathbf{r}_\psi \left( \mathbf{q}_{ij}^C, \chi \right) = \left[ \mathbf{q}_{ij}^{w_{\text{ins}}} \otimes \mathbf{q}_{ij}^{w_{\text{ins}}} \otimes \left( \mathbf{q}_{ij}^C \right) \right]_{xyz}
= \left[ \left( \mathbf{q}_{ij}^{w_{\text{ins}}} \otimes \mathbf{q}_{ij}^b \right)^{-1} \otimes \left( \mathbf{q}_{ij}^{w_{\text{ins}}} \otimes \mathbf{q}_{ij}^b \right) \otimes \left( \mathbf{q}_{ij}^{w_{\text{ins}}} \otimes \mathbf{q}_{ij}^b \right)^{-1} \right]_{xyz},
$$

where $\mathbf{q}_{ij}^{w_{\text{ins}}}$, $\mathbf{q}_{ij}^{w_{\text{orb}}}$, and $\mathbf{q}_{ij}^{w_{\text{orb}}}$ are the quaternion form of $\mathbf{R}_{ij}^{w_{\text{ins}}}$, $\mathbf{R}_{ij}^{w_{\text{ins}}}$, and $\mathbf{R}_{ij}^{w_{\text{orb}}}$, respectively. $\mathbf{R}_{ij}^{w_{\text{ins}}}$ is calculated in Equation (18). The $l$th image is co-viewed with the $j$th image. Lastly, $\psi$ is the set of keyframe pairs in the feature-based map with co-view feature points.

- **Residuals for Scale Estimation in ORB Map**

Since the accuracy of the feature-based map on inter-frame rotation matrix is considerable, the translation requires a scale factor to restore it.

According to the feature-based map and the similarity matrix provided by Umeyama algorithm in the first step, the translation between two co-view frames and scale factor...
where $R_{\text{orb}}^{\text{wvins}}$ is the transformation matrix from the inertial coordinate system of the $j$th image frame to the world coordinate system of the optical flow map. The reason for considering the inertial coordinate system is to further optimize the extrinsic parameters between the camera and the IMU. Lastly, $|\cdot|$ denotes the norm of the vector.

- **Visual Measurement Residual for features tracked by Optical flow**

Consider the $l$th feature that is first observed in the $i$th image, the residual for the feature observation in the $j$th image is defined as:

$$
\mathbf{r}_c(z_{i}^{l}, \chi) = \mathbf{p}_c^{j} - \hat{\mathbf{p}}_c^{j}
$$

$$
\hat{\mathbf{p}}_c^{j} = \pi_c^{-1}\left(\begin{bmatrix} \hat{u}_{i}^{l} \\ \hat{v}_{i}^{l} \end{bmatrix}\right)
$$

$$
\mathbf{p}_c^{j} = \mathbf{R}_b^{c}\left(\mathbf{R}_b^{w}\left(\mathbf{R}_b^{l} \frac{1}{\chi} \pi_c^{-1}\left(\begin{bmatrix} \hat{u}_{i}^{l} \\ \hat{v}_{i}^{l} \end{bmatrix}\right)\right) + \mathbf{P}_w^{l}\right) + \mathbf{P}_w^{w} - \mathbf{p}_b^{b} - \mathbf{f}_c^{l}
$$

where $\begin{bmatrix} \hat{u}_{i}^{l} \\ \hat{v}_{i}^{l} \end{bmatrix}$ is the first observation of the $l$th feature that happens in the $i$th image, and $\begin{bmatrix} \hat{u}_{i}^{l} \\ \hat{v}_{i}^{l} \end{bmatrix}$ is the observation of the same feature in the $j$th image, $\pi_c^{-1}$ is the back projection function, which turns a pixel location into a unit vector using camera intrinsic parameters, $\hat{\mathbf{p}}_c^{j}$ is the unit vector for the observation of the $l$th feature in the $j$th frame, and $\mathbf{p}_c^{j}$ is predicted feature measurement on the unit sphere by transforming its first observation in the $i$th frame to the $j$th frame.

### 3.3.3. Full BA with Some Optimized States Fixed

In the previous step, the inter-frame constraint in the feature-based map is introduced to optimize the state of image frame and extrinsic parameters in optical flow map. The optimized states will be fixed, and then the Full BA operation of visual states in optical flow map and inertial states is performed:

$$
\mathbf{x}_F = \begin{bmatrix}
x_0, & x_1, & \ldots & x_m, & \lambda_0, & \ldots, & \lambda_m, & \chi_l^b
\end{bmatrix}
$$

$$
x_k = \begin{bmatrix}
\mathbf{p}_b^{w}, & \mathbf{v}_b^{w}, & \mathbf{q}_b^{w}, & \mathbf{b}_w, & \mathbf{b}_g
\end{bmatrix},
$$

$$
x_c^b = \begin{bmatrix}
\mathbf{p}_c^b, & \mathbf{q}_c^b
\end{bmatrix}
$$

where $\mathbf{x}_k$ is the IMU state at the time that the $k$th image is captured. It contains position, velocity, and orientation of the IMU in the world frame, and the acceleration bias and gyroscope bias in the IMU body frame.

The visual measurement model here is same as in Local BA, as Equation (24). The only difference is that the position information $(\mathbf{q}_b^{\text{flow}}, \mathbf{P}_b^{\text{flow}})$ of the frame that exits in both the optical flow map and the feature-based map is treated as a constant. To this end, the nonlinear cost function here is formulated as:

$$
\min_{\chi} \left\{ \sum_{(l,j) \in C} \rho \left( \left\| \mathbf{r}_c(z_{i}^{l}, \chi) \right\|_{\mathbf{r}_c^{j}}^2 \right) + \sum_{k \in B} \rho_B \left( \left\| \mathbf{z}_{b+1}^{k} - \mathbf{X}_F \right\|_{\mathbf{p}_b^{k}}^2 \right) \right\},
$$
which is obtained from the feature-based method after scaling to metric unit are added to
\[ r \]
where \( r \) where \( (1) \) MH_01_easy has a fast translational motion in the initialization phase; (2) MH_04_difficult
process based on the optical flow method in VINS-Mono and the feature based method in
proportion in the starting screen; (3) The initialization phase of V1_03_difficult starts with a
pre-integrated IMU measurement terms between two consecutive image frames. Accelerom-
are obtained after several iterations.

\[ \delta \theta \]

Moreover, \( \delta \theta \) is the 3-D error-state representation of quaternion, and \( \delta \hat{b}_k \) are
for the error-state representation, \( \delta \hat{b}_k \) is the 3-D error-state representation of quaternion, and \( \delta \hat{b}_k \) are
pre-integrated IMU measurement terms between two consecutive image frames. Accelerom-
eter and gyroscope biases are also included in the residual terms for the online correction.

4. Experiments and Discussion

To evaluate the effectiveness and practicability of the improved initialization of monoc-
ular visual-inertial SLAM, this paper conducts experiments on the visual-inertial datasets
(EuRoC) collected on-board a micro aerial vehicle (MAV). The datasets contain stereo
images, synchronized IMU measurements, and accurate motion and structure ground-
truth [30]. The main experiments are carried out in three indoor sequences of EuRoC
dataset which were recorded with a micro aerial vehicle equipped with two global-shutter,
monochrome cameras, and an IMU. The two cameras and the IMU were hardware time-
synchronized and were logged at a rate of 20 Hz and 200 Hz, respectively.

The sequences we chose contain the following three cases in the initialization phase:
(1) MH_01_easy has a fast translational motion in the initialization phase; (2) MH_04_difficult
starts from static in the initialization stage, and the weak texture part accounts for a large
stationary phase followed by a fast, large rotation motion. The above three scenarios are
enough to cover most application scenarios. All the experiments are carried out with an
Intel CPU i7-8565U (8 cores @1.8GHz) laptop computer with 16GB RAM.

4.1. Implementation Details

The proposed initialization method is implemented in the combination of VINS-Mono
and ORB-SLAM2. The VINS-Mono is deeply combined with the monocular mode of ORB-
SLAM2 to realize the synchronous input and processing of data. Then, the initialization
process based on the optical flow method in VINS-Mono and the feature based method in
ORB-SLAM2 can be carried out synchronously.

The specific fusion optimization algorithm is shown at Section 3. Firstly, the Umeyama
algorithm is used to estimate the scale of ORB-SLAM2 based on the matching trajectories
obtained by the two initialization threads. Then, the constraints provided by the trajectory
which is obtained from the feature-based method after scaling to metric unit are added to
the modified back-end optimization of VINS-Mono. Finally, the optimized state estimates
are obtained after several iterations.
4.2. Validation of Umeyama Algorithm

In order to verify the effectiveness of the Umeyama algorithm, five tries were performed in the MH_01_easy, MH_04_difficult, and V1_03_difficult sequences, respectively. The comparison between the Umeyama estimated scale values and the actual scale values compared with trajectory truth are shown in Table 1. The scale is determined by the choice of two initialization reference frames, which is different in each time of initialization phase in ORB-SLAM2. The scale to scaling the initialization trajectory of ORB-SLAM2 to metric unit is not the same for each experiment. The ratio in Table 1 represents the scale error between the trajectory of ORB-SLAM2 scaled by the value estimated by Umeyama and the trajectory truth in percentage. It is calculated as Equation (28).

\[ S = \frac{S_{\text{truth}}}{S_{\text{Umeyama}}}, \]  

(28)

From the value of ratio, we can see that the scale estimated by Umeyama is relatively accurate compared with the actual value. To further verify, we compare the ratio value in Table 1 with the ratio value of the ORB-SLAM3 monocular inertial mode trajectory. The ratio value of ORB-SLAM3 is obtained by EVO (Python package for the evaluation of odometry and SLAM) [31]. As shown in Figure 7, the accuracy of the estimated scale is similar to ORB-SLAM3 or even better than ORB-SLAM3. ORB-SLAM3 as the state-of-the-art SLAM system is enough to prove the effect of Umeyama algorithm.

Table 1. Scale Comparision of Umeyama Estimated against the Truth.

| Sequence      | Times | \( S_{\text{Umeyama}} \) | \( S_{\text{truth}} \) | \( S \)   |
|---------------|-------|--------------------------|------------------------|--------|
| MH01_easy     | 1     | 0.7238                   | 0.7246                 | 1.0011 |
|               | 2     | 4.0751                   | 4.1501                 | 1.0184 |
|               | 3     | 4.0546                   | 4.0581                 | 1.0009 |
|               | 4     | 6.2273                   | 6.2763                 | 1.0079 |
|               | 5     | 3.8913                   | 3.9124                 | 1.0054 |
| MH04_difficult| 1     | 9.5073                   | 9.4969                 | 0.9989 |
|               | 2     | 11.046                   | 10.2725                | 0.9300 |
|               | 3     | 8.6863                   | 8.7148                 | 1.0033 |
|               | 4     | 10.0800                  | 9.98360                | 0.9904 |
|               | 5     | 9.5568                   | 9.45228                | 0.9891 |
| V103_difficult| 1     | 2.0469                   | 2.08666                | 1.0194 |
|               | 2     | 3.8243                   | 3.8904                 | 1.0173 |
|               | 3     | 2.2503                   | 2.2093                 | 0.9818 |
|               | 4     | 2.0506                   | 2.0868                 | 1.0177 |
|               | 5     | 2.8825                   | 2.9198                 | 1.0129 |

Figure 7. The distribution of ratio value in five times experiments. The experiments are carried out in MH_01_easy, MH_04_difficult, and V1_03_difficult sequences. The XX_estimate represents the ratio estimated by Umeyama algorithm, and the XX_ORB3VI represents the ratio obtained by ORB-SLAM3 in monocular-inertial mode.
4.3. Evaluation on Optimization Results

The evaluation method is carried out by EVO_APE of EVO, which is mainly used to evaluate the global consistency of the whole trajectory. APE stands for absolute pose error. By default, it executes ATE (absolute trajectory error). To evaluate this error results, we mainly compare from five aspects: max (maximum error), mean (mean value of error), RMSE (root mean square error), SSE (sum of squares for error), and STD (the standard deviation). Each index uses the mean of the five experiments, and the comparison results are shown in Figure 8; (a), (b), and (c) are the comparison results in sequence MH_01_easy, MH_04_difficult, and V1_03_difficult, respectively. It can be found that the optimized trajectory obtained by our proposed method has comparable accuracy with the trajectory obtained by ORB-SLAM3 initialization, while the trajectory obtained by VINS-Mono initialization has poor accuracy. Because VINS-Mono uses the method that the initialization phase completes the coarse estimation of the IMU bias and extrinsic parameters, and then continues to modify these states in the subsequent back-end optimization process, this method cannot guarantee a good positioning accuracy in the initialization phase. In addition, due to the design of parallel threads, an advantage of our proposed method is that even when the extrinsic parameters between the camera and the IMU are unknown, the map construction and tracking process can be started by ORB-SLAM2 in monocular mode, and the scale scaling operation will be realized after the data fusion optimization with the states estimate by initialization process of VINS-Mono.

Figure 8. The comparison of trajectories between our method, VINS-Mono, and ORB-SLAM3 with the trajectory truth in absolute trajectory error, respectively.

In order to show the trajectory comparison of the proposed method more intuitively, we take V1_03_difficult Sequence as an example. The trajectory comparison diagram is shown in Figure 9; (a) is the display of the trajectory in 3D space, and (b) is the error comparison in the XYZ axis. As can be seen from Figure 9, our proposed method is comparable to the monocular-inertial mode of ORB-SLAM3. There are some discontinuities in the trajectory because we only use inertial data in the initialization stage, and the combination algorithm with optical flow method and inertia has not been implemented in the subsequent tracking stage. Therefore, the subsequent trajectory after ORB-SLAM2 was scaled to metric unit, and its tracking process will be lost due to insufficient number of feature points, but it can continue to locate after relocation. The discontinuity of VINS-Mono is because the saved frame pose data is saved together with the key frames in ORB-SLAM2, so the discontinuity also occurs.
The authors would like to thank the anonymous reviewers for their valuable comments on the paper and the builders of the datasets.

Acknowledgments: The authors would like to thank the anonymous reviewers for their valuable comments on the paper and the builders of the datasets.

Conflicts of Interest: The authors declare no conflict of interest.
Appendix A
LK Optical Flow Algorithm

Based on the gray-scale invariance assumption and equality relation in Figure A1, for a pixel located at \((x, y)\) at time \(t\), we suppose that it moves to \((x + dx, y + dy)\) at time \(t + dt\). Due to the gray-scale invariance assumption, there is:

\[
I(x + dx, y + dy, t + dt) = I(x, y, t),
\]

(A1)

The Taylor expansion of the left-hand side is:

\[
(x + dx, y + dy, t + dt) \approx I(x, y, t) + \frac{\partial I}{\partial x} dx + \frac{\partial I}{\partial y} dy + \frac{\partial I}{\partial t} dt,
\]

(A2)

and because of the gray invariant assumption,

\[
\frac{\partial I}{\partial x} dx + \frac{\partial I}{\partial y} dy + \frac{\partial I}{\partial t} dt = 0,
\]

(A3)

divide both sides by \(dt\) and get:

\[
\frac{\partial I}{\partial x} \frac{dx}{dt} + \frac{\partial I}{\partial y} \frac{dy}{dt} = -\frac{\partial I}{\partial t},
\]

(A4)

denoted as \(u\) and \(v\). Moreover, \(\frac{\partial I}{\partial x}\) is the velocity in the \(x\) axis, and \(\frac{\partial I}{\partial y}\) is the velocity in the \(y\) axis, denoted as \((u, v)\). The change of the image gray level with time \(\frac{\partial I}{\partial t}\) is denoted as \(I_t\). Written in matrix form, we have \([I_x \quad I_y] \begin{bmatrix} u \\ v \end{bmatrix} = -I_t\).

Figure A1. LK optical flow algorithm schematic diagram. \(I_i\) denote the image data at time \(t_i\), \(x\) and \(y\) are the pixel coordinates of the corresponding feature point, and \(I(x, y, t)\) represents the gray value of the image frame at pixel coordinates \((x, y)\) at time \(t\).

The above equation is a first-order equation with two variables, and \(u, v\) cannot be solved by the constraint of only one pixel. Therefore, in LK optical flow, assuming that the pixels in a certain window of size \(w \times w\) have the same motion, there is:

\[
[I_x \quad I_y] \begin{bmatrix} u \\ v \end{bmatrix} = -I_{tk}, \quad k = 1, \ldots, w^2,
\]

(A5)

This is an over determined equation with respect to \(u, v\). The traditional solution is to find the least squares solution, which gives the velocity \(u, v\) of the pixel motion between image frames. In SLAM, LK optical flow is commonly used to track the motion of feature points.
29. Lepetit, V.; Moreno-Noguer, F.; Fua, P. EPnP: An accurate O(n) solution to the PnP problem. *Int. J. Comput. Vis.* **2009**, *81*, 155–166. [CrossRef]

30. European Robotics Challenge. Available online: http://http://www.euroc-project.eu/ (accessed on 1 June 2021).

31. Evo. Available online: https://github.com/MichaelGrupp/evo (accessed on 1 June 2021).