Hindawi
Journal of Sensors
Volume 2021, Article ID 5565158, 18 pages
https://doi.org/10.1155/2021/5565158

Research Article

Fast Monocular Visual-Inertial Initialization with an Improved Iterative Strategy

Jun Cheng, Liyan Zhang, and Qihong Chen

School of Automation, Wuhan University of Technology, Wuhan 430070, China

Correspondence should be addressed to Liyan Zhang; zlywhut@whut.edu.cn

Received 13 January 2021; Revised 11 March 2021; Accepted 24 April 2021; Published 28 May 2021

Academic Editor: José A. Padilla-Medina

The initialization process has a great effect on the performance of the monocular visual inertial simultaneous localization and mapping (VI-SLAM) system. The initial estimation is usually solved by least squares such as the Gauss-Newton (G-N) algorithm, but the large iteration increment might lead to the slow convergence or even divergence. In order to solve this problem, an improved iterative strategy for initial estimation is proposed. The methodology of our initialization can be divided into four steps: Firstly, the pure visual ORB-SLAM model is utilized to make all variables observable. Secondly, the IMU preintegration technology is adopted for IMU-camera frequency alignment at the same time with key frame generation. Thirdly, an improved iterative strategy which is based on the trust region is introduced for the gyroscope bias estimation as well as the gravity direction is refined. Finally, the accelerometer bias and visual scale are estimated on the basis of previous estimations. Experimental results on the public datasets show that the estimation of initial values can be converged faster, as well as the velocity and pose of sensor suite can be estimated more accurately than the original method.

1. Introduction

With the development of artificial intelligent (AI), the monocular visual inertial simultaneous localization and mapping (VI-SLAM) technology has become an active research topic in the robotics and computer vision communities. The camera image contains a rich information of the surrounding environment, in which it can be utilized to estimate the camera poses, as well as restructure the sparse or the dense maps. The IMUs provide the measurements of angular velocity and local linear acceleration, which can be utilized to estimate the rigid body motion in a short period. The complementary features make the visual-inertial combination suitable for many applications such as autonomous or semiautonomous driving [1, 2], unmanned aerial robots [3, 4], augmented reality (AR) [5, 6], and 3D reconstruction [7, 8]. At present, the tightly coupled nonlinear optimization method is widely applied to the visual/visual-inertial SLAM, such as ORB-SLAM and ORBSLAM 2/3 [9–11], OKVIS [12], VINS-MONO/VINS-FUSION [13, 14], VI-DSO [15], and VI-ORBSLAM [16]. The estimation of initial values has a great effect on the aforementioned VI-SLAM system. Specifically, the IMU's initial value estimation is of great significance to the initialization process; once these parameters are obtained successfully, the measurements of the inertial measurement unit (IMU) can be used to improve the accuracy and robustness of continuous tracking, as well as to find the metric scale of the three-dimensional (3D) visual map.

In early studies, several initialization methods have been studied, such as the representative joint methods [17–19] and disjoint methods [16, 20, 21].

(i) The joint visual-inertial initialization method is pioneered by Martinelli [17], which assumes that all features are correctly tracked in all frames. The spurious tracks lead to a poor real-time performance. In later research, it is improved by Kaiser et al. with a little bit of precision that is sacrificed [18]. The method in [19] suffers a low initialization recall; it only works in twenty percent of the trajectory points, which might be a problem for the robot applications in case the system needs to be launched immediately

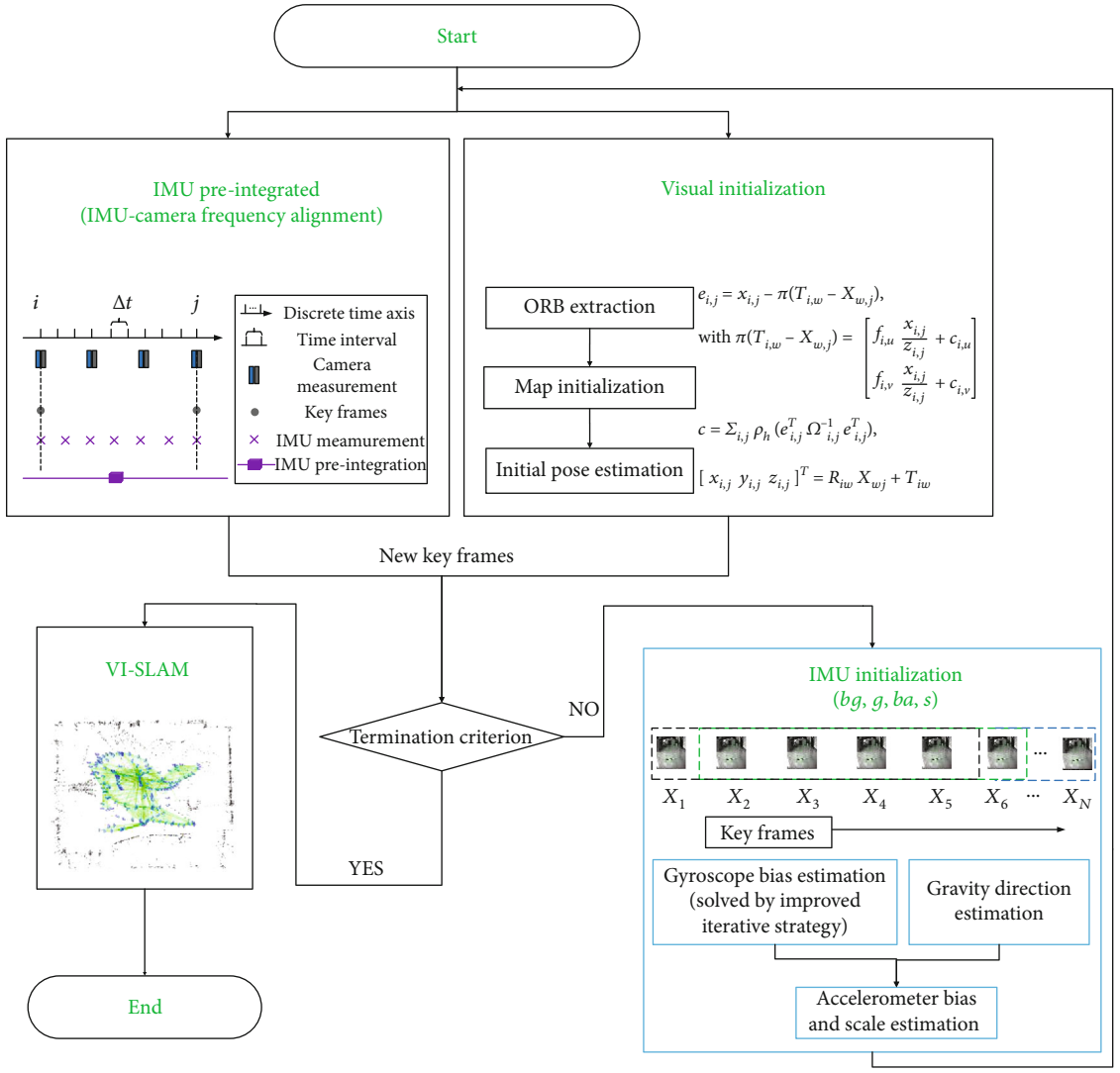


Figure 1: Flowchart of visual-inertial initialization. The process of IMU initialization is highlighted by a blue box. Abbreviations: bg: gyroscope bias; g: gravity; ba: accelerometer bias; s: visual scale; v: velocity.

(ii) The disjoint method is first introduced by Murartal and Tardos [16] and latter adapted by Qin and Shen and Yang and Shen [20, 21] with a good performance. In both cases, the parameters of IMU are estimated in different steps by solving a series of linear formulas with the least-squares method such as Gauss-Newton (G-N) and Levenberg-Marquardt (L-M) [22, 23]. The G-N method is built on the basis of the Newton method. It performs a high local convergence speed, but several limitations still exist such as the large iteration increment which may lead to the slow global convergence speed [24, 25]. In the later research, Levenberg [26] and Marquardt [27] suggest to use a damped G-N method, in which the size and direction of the iterative step are influenced by the damped parameter. It makes this method without a specific line search which guarantees the global convergence performance [28]. However, solving a set of complexity equations needs several times for each iteration, which may lead to a reduction in the speed of solution

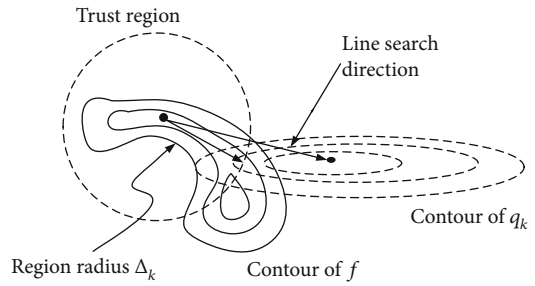


Figure 2: The model of truth region-based iterative strategy.

To sum up, the speed of initialization convergence is important for the VI-SLAM system. The contribution of this paper is that an improved iterative strategy is proposed based on the trust region method to speed up the initial estimation.

The flowchart of our initialization method is shown in Figure 1. It can be divided into four steps: firstly, the pure visual ORB-SLAM model is adopted to make all variables

Inputs: Δ_k,ω_k,$\tilde{\gamma}_k$
Output: Δ_{k+1}
Step 1: If $\omega_k > \mu_2$, then put:$\Delta_{k+1} \leftarrow \max\{\Delta_k, c_2\|\tilde{\gamma}_k\|\}$where $0 < c_1 \le c_2, 0 < \mu_1 < \mu_2 < 1$
Step 2: Else if $\omega_k < \mu_1$,then put: $\Delta_{k+1} \leftarrow c_1\Delta_k$

Algorithm 1: Updating region radius Δ_k.

Inputs: Δ_k,γ_k^{cd},γ_k^{gn}
Output: $\tilde{\gamma}_k$
Step 1: If $\|\gamma_k^{cd}\| \ge \Delta_k$,then output: $\tilde{\gamma}_k \leftarrow (\Delta_k/\|\gamma_k^{cd}\|)\gamma_k^{cd}$
Step 2: Else if $\|\gamma_k^{gn}\| \le \Delta_k$, then output: $\tilde{\gamma}_k \leftarrow \gamma_k^{gn}$
Step 3: Else output: $\tilde{\gamma}_k \leftarrow \gamma_k^{cd} + (\beta - 1)(\gamma_k^{gn} - \gamma_k^{cd})$, where β is calculated by equation (17).

Algorithm 2: Computing the step $\tilde{\gamma}_k$.

observable in the preliminary stage. Secondly, the IMU preintegration technology is adopted for IMU-camera frequency alignment at the same time with key frame generation. Thirdly, an iterative strategy which is based on the trust region is introduced for the gyroscope bias estimation while the gravity direction is refined. Finally, the accelerometer bias and visual scale are estimated on the basis of previous estimation. In Experiments, qualitative and quantitative analyses on the public datasets [29] are given to demonstrate our improved effect. The results show that the estimation of initial values can be converged in a faster speed, as well as the velocity and poses of a sensor suite can be estimated more accurately than the original method.

The remainder part of this paper is organized as follows: Section 2 describes the process of IMU initial estimation. Then, the improved iterative strategy is described in Section 3. The experiments are described in Section 4. The conclusion is drawn in Section 5.

2. IMU Initial Estimation

In this section, the initial parameters of IMU are estimated. The relationships of IMU body frame {B} and camera frame {C} are defined with the scale factor **s** taken into account; it is described as follows:

$$\begin{aligned} \mathbf{R}_{\mathrm{WB}} &= \mathbf{R}_{\mathrm{WC}} \cdot \mathbf{R}_{\mathrm{CB}}, \\ {}_{W}\mathbf{P}_{\mathrm{B}} &= \mathbf{R}_{\mathrm{WC}} \cdot {}_{C}\mathbf{P}_{\mathrm{B}} + \mathrm{s} \cdot {}_{W}\mathbf{P}_{\mathrm{C}}, \end{aligned} \tag{1}$$

where **R** and **P** represent rotation and translation vector, respectively.

The IMU preintegration technology [30, 31] is adopted for IMU-camera frequency alignment. In order to make all variables observable, the pure monocular visual SLAM system [9, 10] is launched to work a few seconds for generating key frames. The detailed process of the IMU parameter estimation is described as follows.

2.1. Estimating Gyroscope Bias. The gyroscope bias estimation could be obtained from the known orientation of two consecutive key frames. Firstly, we assume that the change of bias is negligible; i.e., the bias b_{g} is a tiny constant value. Then, the difference between the preintegration of gyroscope measurements and the orientation estimated by pure visual SLAM is minimized:

$$\arg\min_{b_{\mathrm{g}}} \sum_{i=1}^{N-1} \left\| \mathrm{Log}\left(\Delta\mathbf{R}_{i,i+1}\mathrm{Exp}\left(\mathbf{J}_{\Delta R}^{\mathrm{g}}\mathbf{b}_{\mathrm{g}}\right)\right)^{T} \mathbf{R}_{\mathrm{BW}}^{i+1}\mathbf{R}_{\mathrm{WB}}^{i} \right\|^2, \tag{2}$$

where N denotes the total quantity of key frames, $\mathbf{R}_{\mathrm{WB}}^{(.)} = \mathbf{R}_{\mathrm{WC}}^{(.)} \times \mathbf{R}_{\mathrm{CB}}$ is computed from the orientation $\mathbf{R}_{\mathrm{WC}}^{(.)}$ and the calibration $\mathbf{R}_{\mathrm{CB}}$,$\Delta\mathbf{R}_{i,i+1}$ denotes the preintegration of gyroscope measurements between two consecutive key frames, $\mathrm{Exp}(.)$ denotes the exponential map with $R = \mathrm{Exp}(\phi) = \exp(\phi^{\Lambda})$, where R denotes a rotation matrix, $R \to SO(3)$, ϕ is a corresponding vector, $\phi \to so(3)$, and $\mathbf{J}_{\Delta R}^{\mathrm{g}}$ represents the Jacobian matrix. Analytic Jacobian matrices for a similar expression can be found in [31].

2.2. Estimating Gravity Direction. The gravity direction has a great influence on the estimation of acceleration; it should be refined before the parameter estimation of accelerometer bias and scale. In particular, a new constraint, i.e., gravity magnitude G ($G \approx 9.8$), is introduced. In terms of the frame {W}, the gravity direction can be computed as follows:

$$\bar{g}_{\mathrm{w}} = \frac{g_{\mathrm{W}}^{*}}{\|g_{\mathrm{W}}^{*}\|}. \tag{3}$$

The rotation $\mathbf{R}_{\mathrm{WI}}$ can be calculated from the angle θ between the two direction vectors:

$$\mathbf{R}_{WI} = E\mathrm{xp}(\bar{v}\theta). \tag{4}$$

With $\bar{v} = (\bar{g}_{\mathrm{I}} \times \bar{g}_{\mathrm{W}})/\|\bar{g}_{\mathrm{I}} \times \bar{g}_{\mathrm{W}}\|, \theta = a\tan 2(\|\bar{g}_{\mathrm{I}} \times \bar{g}_{\mathrm{W}}\|, \bar{g}_{\mathrm{I}} \cdot \bar{g}_{\mathrm{W}})$, the gravity vector can be expressed as follows:

$$\mathbf{g}_{\mathrm{W}} = \mathbf{R}_{\mathrm{WI}}\bar{g}_{\mathrm{I}}G \tag{5}$$

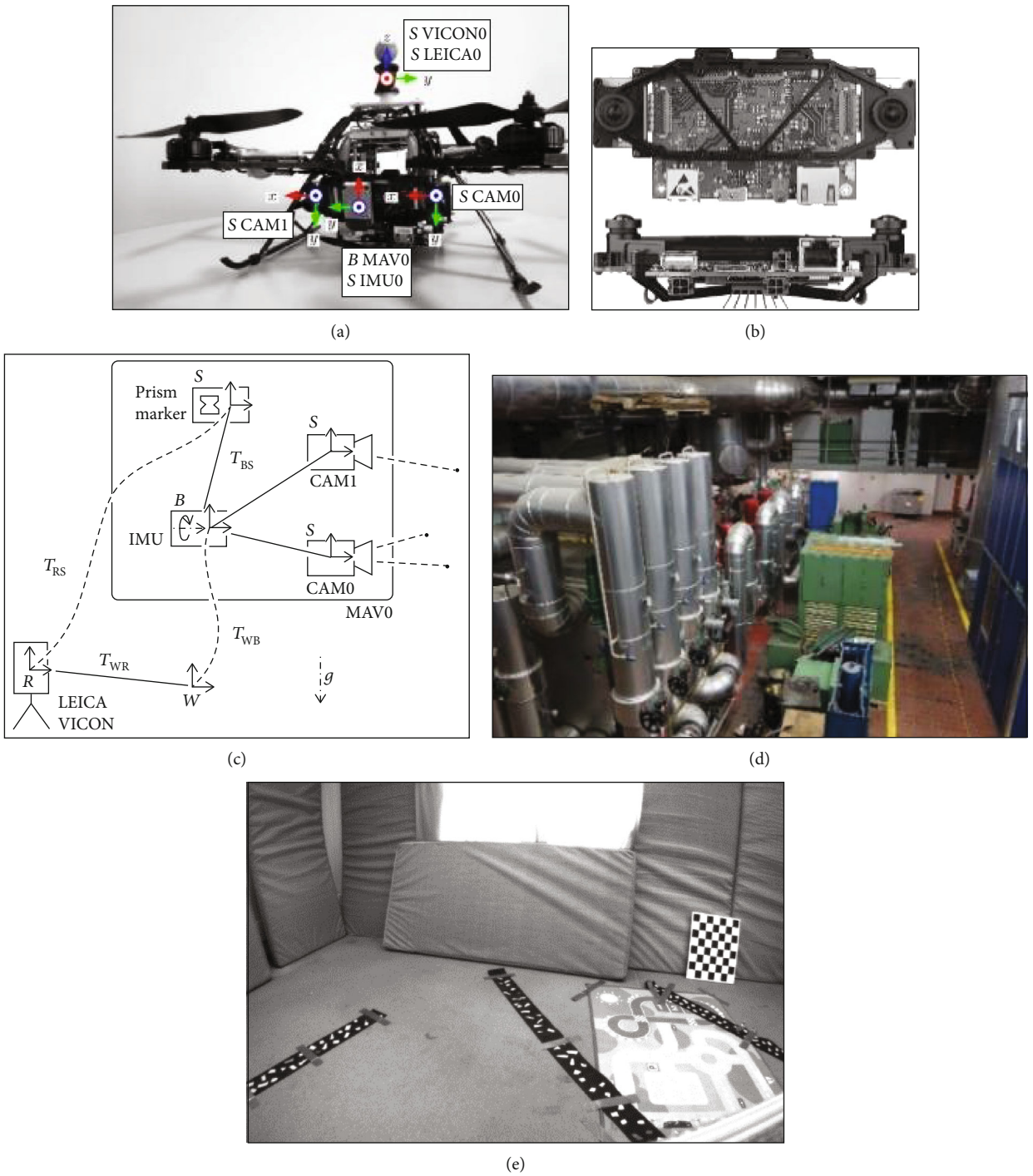
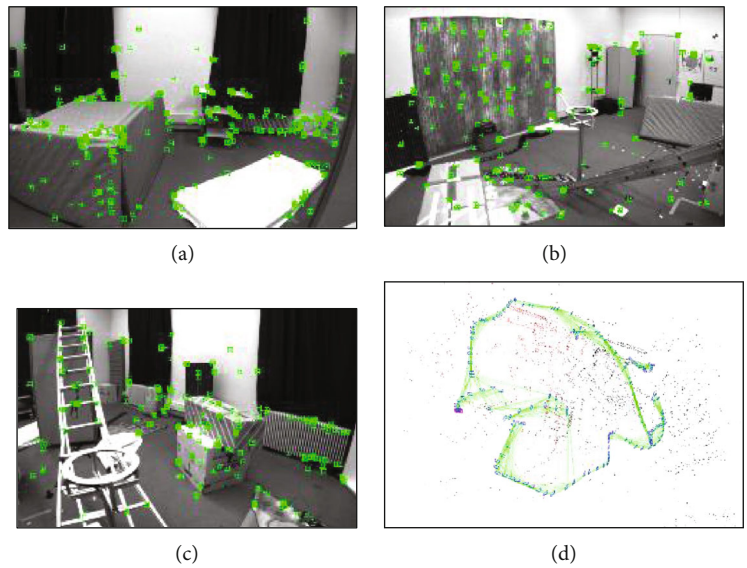


FIGURE 3: The platform and scenes of the EuRoC dataset: (a) the micro aerial vehicle assembled two front cameras, an IMU sensor and a motion capture system; (b) the VI sensor suite; (c) the frame of the system; (d, e) the scenes of the machine hall and man-made laboratory room, respective.

where $\mathbf{R}_{\mathrm{WI}}$ can be calculated by just two angles around the x axis and y axis in frame {I} and the rotation around the z axis has no effect in $\mathbf{g}_{\mathrm{W}}$.

2.3. Estimating Accelerometer Bias and Scale. Once the accurate gyroscope bias and gravity vector are obtained, the positions, velocities, and rotation can be calculated by the integral operation. Considering the influence caused by the accelerometer bias, the $\mathbf{R}_{\mathrm{WI}}$ is also adjusted, where it can be expressed by a two degree of freedom disturbance $\delta\theta$; Equation (5) can be rewritten as follows:

$$\begin{aligned}\mathbf{g}_{\mathrm{W}} &= \mathbf{R}_{\mathrm{WI}}\mathrm{Exp}(\delta\theta)\bar{g}_{\mathrm{I}}G \approx \mathbf{R}_{\mathrm{WI}}\bar{g}_{\mathrm{I}}G \\ &\quad + \mathbf{R}_{\mathrm{WI}}(\delta\theta)^{\wedge}\bar{g}_{\mathrm{I}}G = \mathbf{R}_{\mathrm{WI}}\bar{g}_{\mathrm{I}}G - \mathbf{R}_{\mathrm{WI}}(\bar{g}_{\mathrm{I}})^{\wedge}G\delta\theta,\end{aligned} \tag{6}$$



(a) (b) (c) (d)

FIGURE 4: The sparse mapping results on *V2_01_easy* sequence: (a–c) detected point features for different frames; (d) the reconstructed point map (black: previous map points, red: current map points) with the key frame trajectory (blue).

with $\delta\theta = [\delta\theta_{xy}^T, 0]^T$, $\delta\theta_{xy} = [\delta\theta_x, \delta\theta_y]^T$. Considering the effect of accelerometer bias, it can be obtained that

$$\begin{aligned} s_W\mathbf{p}_C^{i+1} = {} & s_W\mathbf{p}_C^i + {}_W\mathbf{v}_B^i\Delta t_{i,i+1} - \frac{1}{2}\mathbf{R}_{WI}(\bar{g}_I)\times G\Delta t_{i,i+1}^2\delta\theta \\ & + \mathbf{R}_{WB}^i\left(\Delta p_{i,i+1} + J_{\Delta p}^a b_a\right) + \left(\mathbf{R}_{WC}^i - \mathbf{R}_{WC}^{i+1}\right){}_C\mathbf{p}_B \\ & + \frac{1}{2}\mathbf{R}_{WI}\bar{g}_I G\Delta t_{i,i+1}^2. \end{aligned} \tag{7}$$

The velocities can be eliminated when the constraints between three consecutive key frames are taken into account. The linear relationship is obtained as follows:

$$[\lambda(i)\varphi(i)\zeta(i)]\begin{bmatrix} s \\ \delta\theta_{xy} \\ b_a \end{bmatrix} = \psi(i), \tag{8}$$

where $\lambda_{(i)}$, $\varphi(i)$, $\zeta(i)$, and $\psi(i)$ are parameterized as follows:

$$\begin{aligned} \lambda(i) &= \left({}_W\mathbf{p}_C^2 - {}_W\mathbf{p}_C^1\right)\Delta t_{23} - \left({}_W\mathbf{p}_C^3 - {}_W\mathbf{p}_C^2\right)\Delta t_{12}, \\ \varphi(i) &= \left[\frac{1}{2}\mathbf{R}_{WI}(\bar{g}_I)\times G\left(\Delta t_{12}^2\Delta t_{23} + \Delta t_{23}^2\Delta t_{12}\right)\right]_{(:,1:2)}, \\ \zeta(i) &= \mathbf{R}_{WB}^2 J_{\Delta p23}^a\Delta t_{12} + \mathbf{R}_{WB}^1 J_{\Delta v23}^a\Delta t_{12}\Delta t_{23} - \mathbf{R}_{WB}^1 J_{\Delta p12}^a\Delta t_{23}, \\ \psi(i) &= \left(\mathbf{R}_{WC}^2 - \mathbf{R}_{WC}^1\right){}_C\mathbf{p}_B\Delta t_{23} - \left(\mathbf{R}_{WC}^3 - \mathbf{R}_{WC}^2\right){}_C\mathbf{p}_B\Delta t_{12} \\ & \quad + \mathbf{R}_{WB}^2\Delta p_{23}\Delta t_{12} + \mathbf{R}_{WB}^1\Delta v_{12}\Delta t_{12}\Delta t_{23} \\ & \quad - \mathbf{R}_{WB}^1\Delta p_{12}\Delta t_{23} + \frac{1}{2}\mathbf{R}_{WI}\bar{g}_I G\Delta t_{ij}^2, \end{aligned} \tag{9}$$

where $[]_{(:,1:2)}$ denotes the first two columns of the matrix. Stacking all relations between three consecutive key frames in (8), a linear system can be formed as the following equation $\mathbf{A}_{3(N-2)\times 6}\mathbf{X}_{6\times 1} = \mathbf{B}_{3(N-2)\times 1}$. It can be solved by the singular value decomposition (SVD) method. In this case, it contains $3(N-2)$ equations and 6 unknown variables, and at least 4 key frames are needed to solve the system.

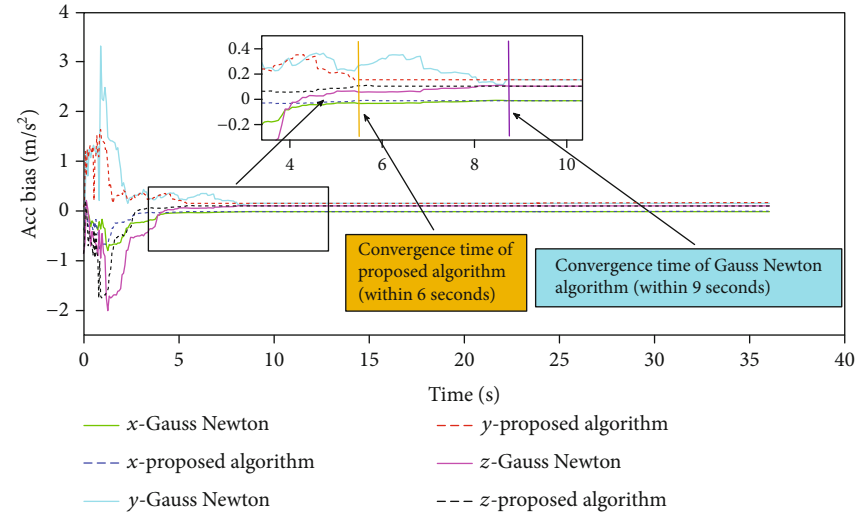
3. Improved Iterative Strategy

As Equation (2) is a typical nonlinear least-square problem, the common solving method is the G-N algorithm which is adopted in [16]. The G-N method provides a high local convergence speed, but the large iteration increment might lead to the slow global convergence or even divergence. To tackle this problem, an improved iterative strategy is introduced. In particular, our method is a trust region-based method which combines the steepest descent and G-N algorithms; the model of truth region is shown in Figure 2.

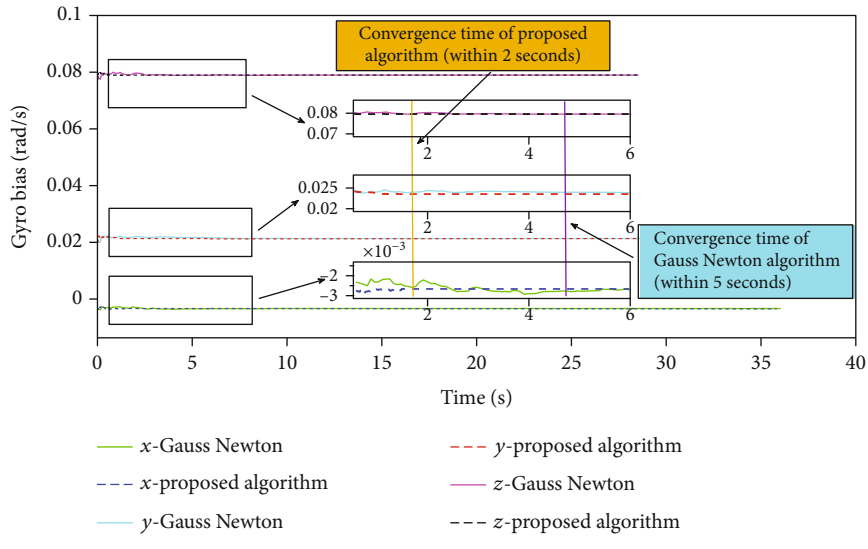
Similar to the L-M algorithm, the objective function is approximated with a trust model, while the solution is considered as the minimum of model function around the current point. Then, the following minimization subproblem is solved in each iteration step:

$$\min\left\{q_k(\gamma) = f_k + g_k^T\gamma + \frac{1}{2}\gamma^T G_k\gamma \mid \gamma\in\Gamma_k, \quad \|\gamma\| \le \Delta_k\right\}, \tag{10}$$

where $f_k = f(x_k)$ is the objective function, Equation (10) is the approximate model of f_k around the current point x_k, $g_k = \nabla f(x_k)$ is the gradient of f_k, Δ_k represents the radius of the trust region, $\gamma = x_{k+1} - x_k$, Γ_k is the class of path, $G_k = \nabla^2 f(x_k)$ is the Hessian matrix of $f(x_k)$, and $\|.\|$ denotes the



(a)



(b)

Figure 5: Continued.

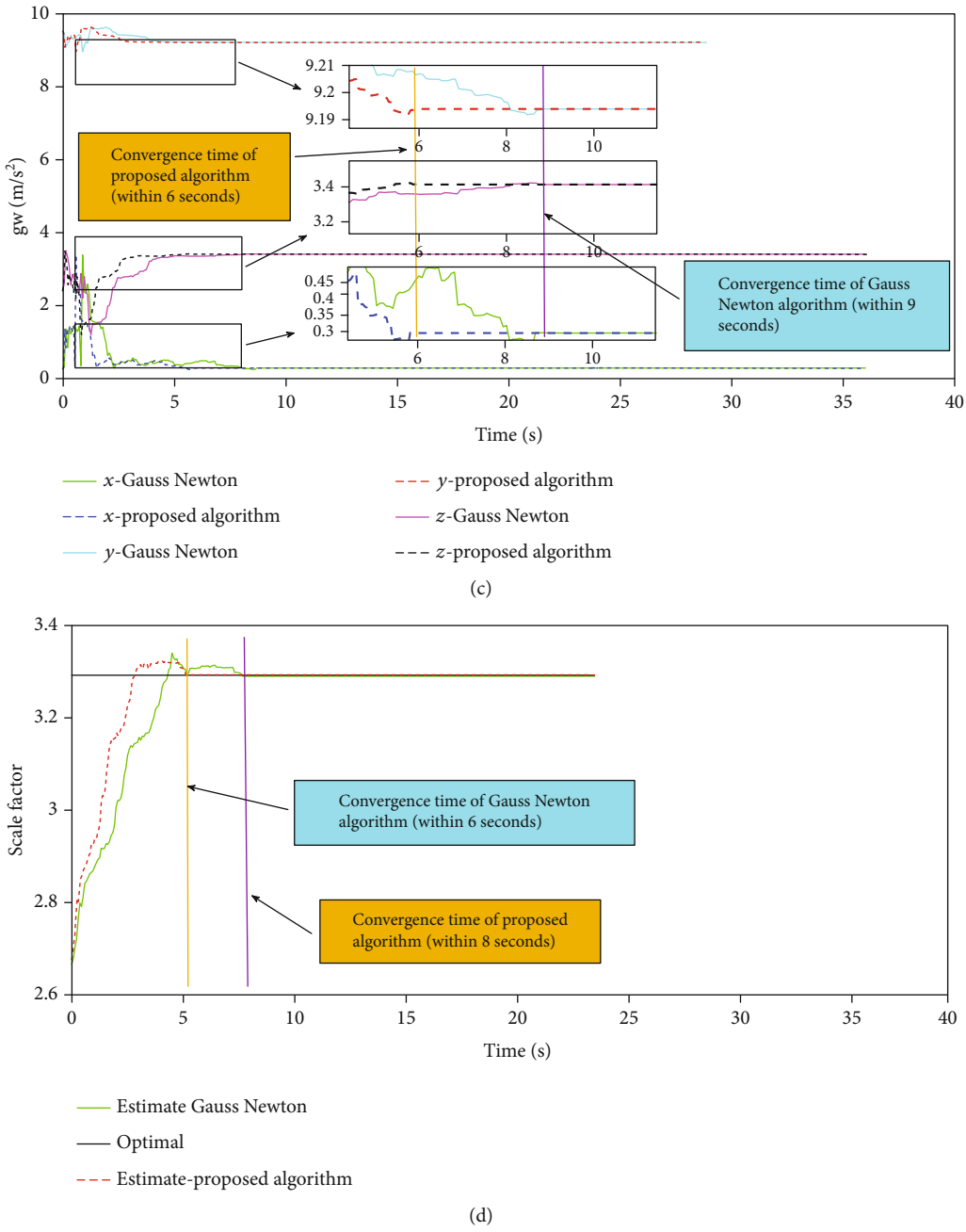


FIGURE 5: The time-varied characteristic curves of the initial estimations: (a) accelerometer bias characteristic curves; (b) gyroscope bias characteristic curves; (c) gravity characteristic curves; (d) visual scale characteristic curves. Abbreviations: {x-gn, y-gn, z-gn} and {x-our, y-our, z-our} represent the estimation of G-N and our algorithms with x, y, and z directions, respectively.

norm. At each iteration step, the radius of the trust region is adjusted with ω_k changed:

$$\omega_k = \frac{f(x_k) - f(x_k + \gamma_k)}{\eta_k(0) - \eta_k(\gamma_k)}, \tag{11}$$

whereω_k is defined as the gain ratio, the denominator: $\eta_k(0) - \eta_k(\gamma_k)$, and numerator: $f(x_k) - f(x_k + \gamma_k)$, is the predicted and actual reduction values, respectively. The steps of updating Δ_k are described in Algorithm 1.

In this work, the sufficient reduction condition for global convergence is assumed that the reduction of obtained step performs at least square to the reduction of Cauchy step [32], which is described as follows:

$$\gamma_k^{\mathrm{cd}} = -\frac{g_k^T g_k}{g_k^T J_x^T J_x g_k} g_k, \tag{12}$$

where J_x represents the Jacobian matrix; it includes the gradient of residuals which can be found in [22].

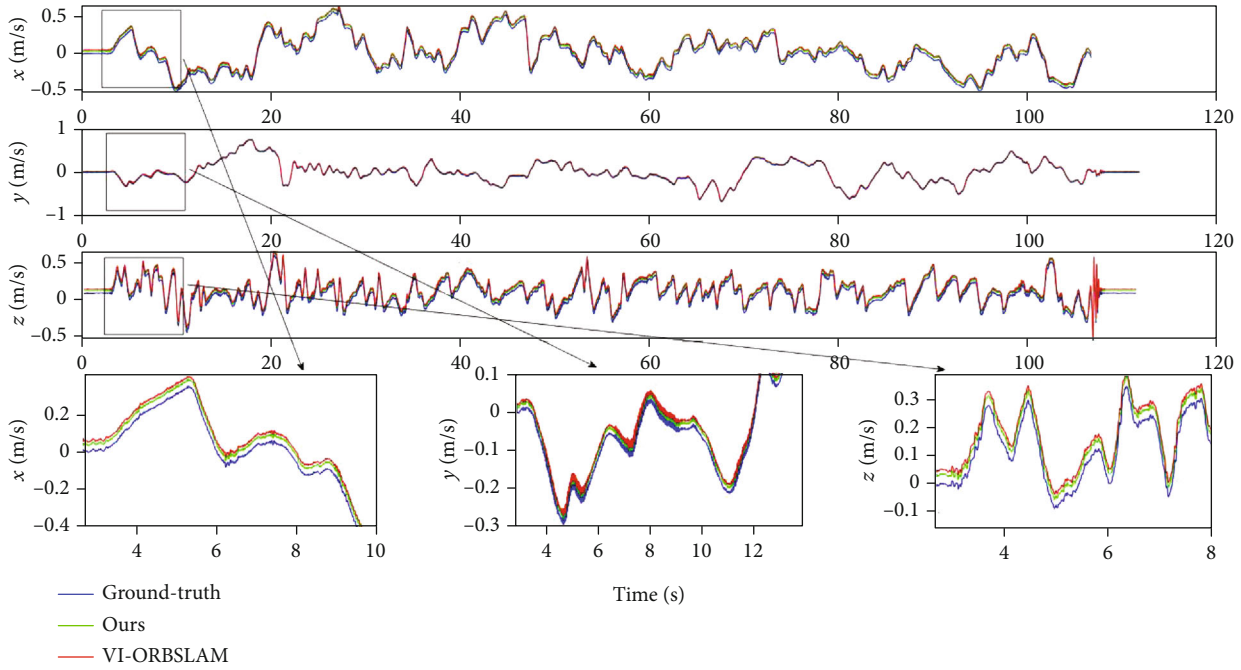


Figure 6: Velocity estimation in the x, y, and z (m/s) directions of the V2_01_easy sequence. Blue: velocity of ground truth, red: velocity estimation of VI-ORBSLAM with the Gauss-Newton method used in initialization process, and green: velocity estimation of ours which uses the improved iteration method in the initialization process.

Table 1: The quantified velocity drift of whole path on *V2_01* sequence. All the results are the median over 10 tests. Abbreviations: med: median; dirt: direction.

	VI-ORBSLAM med drift (m/s)	Ours med drift (m/s)	Improvement (%)
X_{dirt}	0.0293	0.0195	33.45
Y_{dirt}	0.0165	0.0110	33.33
Z_{dirt}	0.0501	0.0385	23.15

Assuming that $\gamma_k^*(\Delta)$ is the solution of Equation (10), due to our method which combines the steepest descent and G-N algorithms, the solution $\gamma_k^*(\Delta)$ is obtained on the basis of the selection of one of the two steps γ_k^{cd} and γ_k^{gn}. The relation of the two steps is described as follows:

$$\begin{aligned} \left\|\gamma_k^{\text{cd}}\right\| &\le \left\|\gamma_k^{\text{gn}}\right\|, \\ \eta\left(\gamma_k^{\text{gn}}\right) &\le \eta\left(\gamma_k^{\text{cd}}\right), \end{aligned} \tag{13}$$

where γ_k^{cd} is the step of the steepest descent algorithm and γ_k^{gn} is the step of the G-N algorithm.

According to Equation (13), there are two states occur for γ_k^{cd} and γ_k^{gn}:

(1) The G-N point is outside of the trust region

The path of our iterative strategy consists of two conditions in which the first case is a line segment starting from the current point to γ_k^{cd} and the second case is a line segment extending from γ_k^{cd} to γ_k^{gn}. Specifically, it can be formulated by $\tilde{\gamma}_k(\beta), \beta \in [0, 2]$:

$$\tilde{\gamma}_k(\beta) = \begin{cases} \beta\gamma_k^{\text{cd}}, & 0 \le \beta < 1, \\ \gamma_k^{\text{cd}} + (\beta - 1)\left(\gamma_k^{\text{gn}} - \gamma_k^{\text{cd}}\right), & 1 \le \beta < 2, \end{cases} \tag{14}$$

where β can be computed from the following equation:

$$\left\|\gamma_k^{\text{cd}} + (\beta\text{-}1)\left(\gamma_k^{\text{gn}} \text{-} \gamma_k^{\text{cd}}\right)\right\|^2 = \Delta_k^2. \tag{15}$$

(2) The G-N point is inside of the trust region

$$\gamma_k^*(\Delta) = \gamma_k^{\text{gn}}, \quad \text{while } \left\|\gamma_k^{\text{gn}}\right\| \le \Delta_k. \tag{16}$$

Thus, the solution of our method is obtained on the approximate path with the minimum point of model function; the processes are shown in (Algorithm 2).

4. Experiments

In this section, the proposed initialization algorithm is integrated into the VI-ORBSLAM framework [16], and several tests are evaluated on the EuRoC dataset [29]. In order to display the excellent performance, we compare our method with the original VI-ORBSLAM, VINS-MONO, and the monocular version of ORB-SLAM3 frameworks.

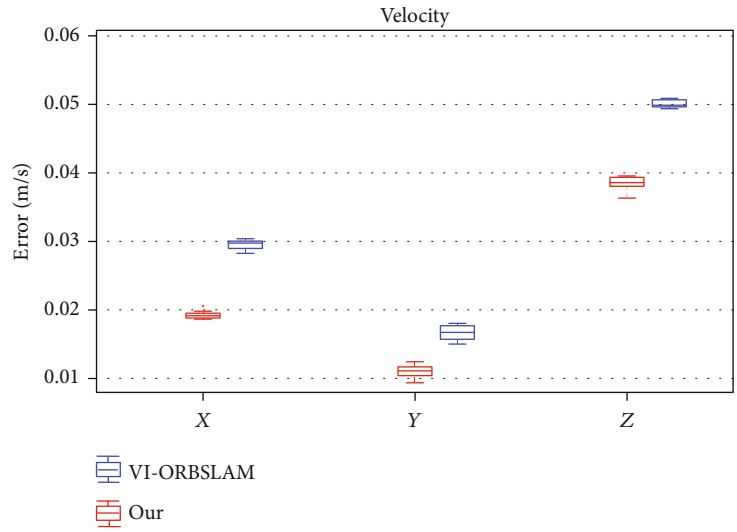


Figure 7: The boxplot of the velocity errors in the x, y, and z axes. The blue color represents the result of VI-ORBSLAM; the red color represents the result of our method.

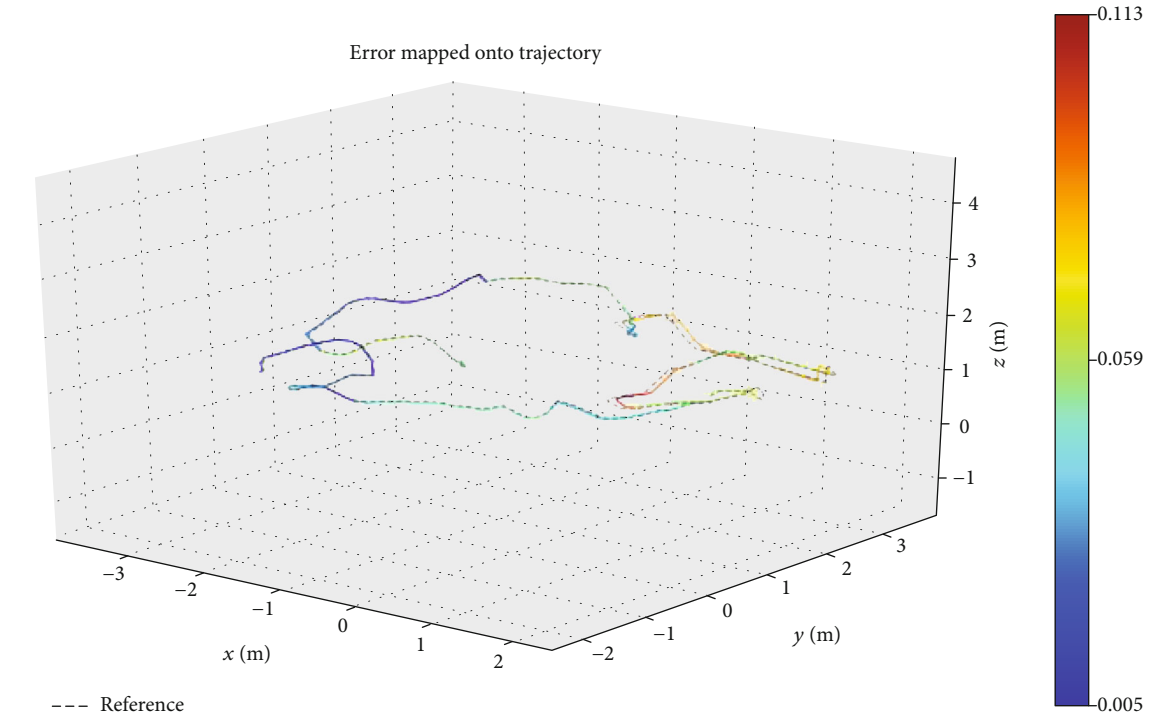
4.1. EuRoC Dataset. The EuRoC dataset with 11 sequences is collected by a micro aerial vehicle (MAV) platform with a visual inertial sensor suite assembled. The environment of data collection is shown in Figure 3, which consists of machine hall scene and man-made laboratory room.

According to the texture, illumination, motion blur, and fast/slow motions, the sequences are classified into easy, medium, and difficult sets [33]. The binocular cameras (denoted as CAM0 and CAM1) and the IMU are logged at 20 and 200 Hz with hardware time-synchronized, respectively. Besides, the dataset provides not only accurate ground truth of the moving trajectories but also the biases of accelerometer and gyroscope, and the velocities of IMU body are provided. In this work, we only use the monocular (left camera) and the IMU measurements. All experiments are carried out on the computer with 16GB RAM, i7-9700 CPU (8 cores @3.00GHz), in Ubuntu 18.04+ Melodic operating system. For more convenient analysis, the performance of parameter convergence, velocity estimation, and the localization accuracy is evaluated by using *V2_01_easy* sequence of the EuRoC dataset, in which the accelerometer bias and gyroscope bias are approximately [-0.0236, 0.1210, 0.0748] m/s^2 and [-0.0023, 0.0250, 0.0817] rad/s, respectively. The sparse mapping results on *V2_01_easy* sequence with the key frame trajectory are shown in Figure 4. The convergence performance of the initial parameter, velocity estimation errors, and key frame absolute trajectory error (ATE) of the whole 11 sequences is also analyzed. The detailed description is shown as follows.

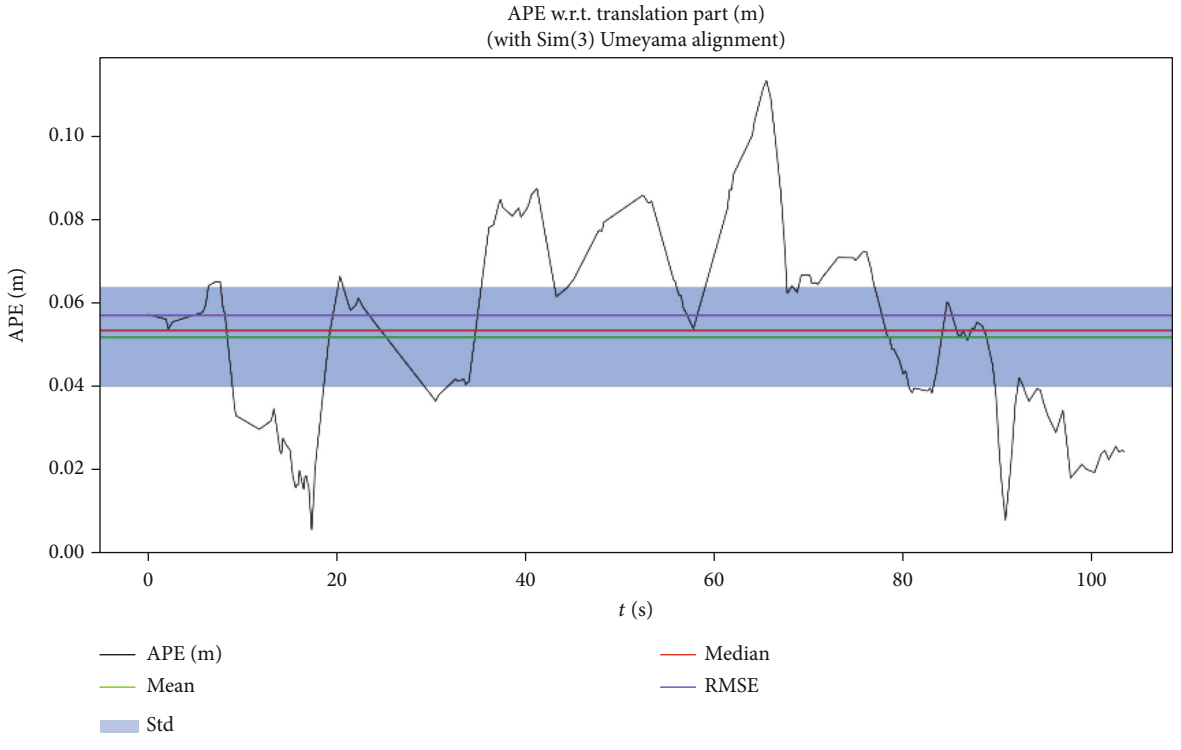
4.2. Initialization Parameter Convergence. The time-varied characteristic curves of the initial estimations: accelerometer bias $\mathbf{b}_a$, gyroscope bias $\mathbf{b}_g$, gravity $\mathbf{g}_W$, and visual scale $\mathbf{s}$, are shown in Figure 5. We compare the convergence performance of the two optimization-based methods: the Gauss-Newton method and the proposed method. It can be seen that all parameters converge successfully in these two methods; the Gauss-Newton- (G-N-) based method converges within 9 seconds, while our based method converges within 6 seconds. In particular, as shown in Figures 5(a) and 5(c), the characteristic curves of accelerometer bias and gravity encounter severe oscillation within 4 seconds. This is because the MAV platform does not have enough excitation on the sensor suite in the stage of stationary and slight disturbance, which makes the accelerometer bias and the gravity vector hard to distinguish. Besides, the estimation of gyroscope bias converges to stable values in a very short time (within 2 seconds of ours) which is shown in Figure 5(b).

It well confirms that the proposed iterative strategy described in Section 3 achieves good performance. The estimation of visual scale is plotted in Figure 5(d), it should be noted that the optimal value of scale factor is obtained by manually scaling and aligning the estimated IMU body trajectories to the ground-truth trajectories in advance. The result shows that our method also converges to the optimal value with a faster speed than the G-N-based method.

4.3. Velocity Estimation. The time characteristic curves of the estimated velocity with x, y, and z directions are shown in Figure 6. It draws the comparison results of VI-ORBSLAM (red line), ground truth (blue line), and our proposed algorithm (green line) which are tested on the *V2_01 sequence* of the EuRoC dataset. Due to the estimations and ground truth that are expressed in different reference frames, the estimated velocities need to be aligned with the ground truth in advance. It can be known that the velocity estimation suffers different drifts in x, y, and z directions, but the error of ours is smaller than that of VI-ORBSLAM. The velocity drift of the whole path is quantified in Table 1. Compared with the velocity drift of VI-ORBSLAM: [0.0293,0.0165,0.0501] m/s, the drift of ours is [0.0195,0.0110,0.0385] m/s in the x, y, and z axes; it can be calculated that the accuracy of ours is improved by [33.45,33.33,23.15] %. The percentage improvement in accuracy can be calculated as $|\text{Error}^* - \text{Error}^{\oplus}|/|\text{Error}^{\oplus}| * 100\%$, where $\text{Error}^{\oplus}$ and Error^* indicate the drift of VI-ORBSLAM and the drift of our method, respectively.

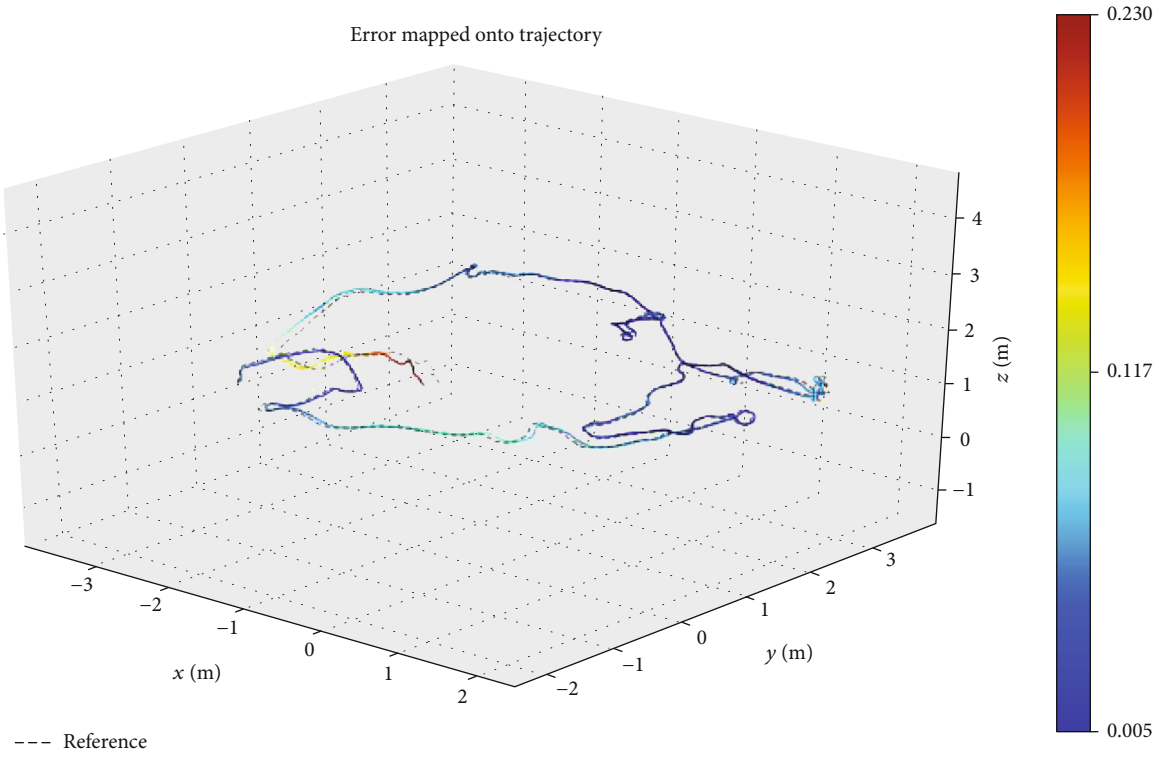


(a)

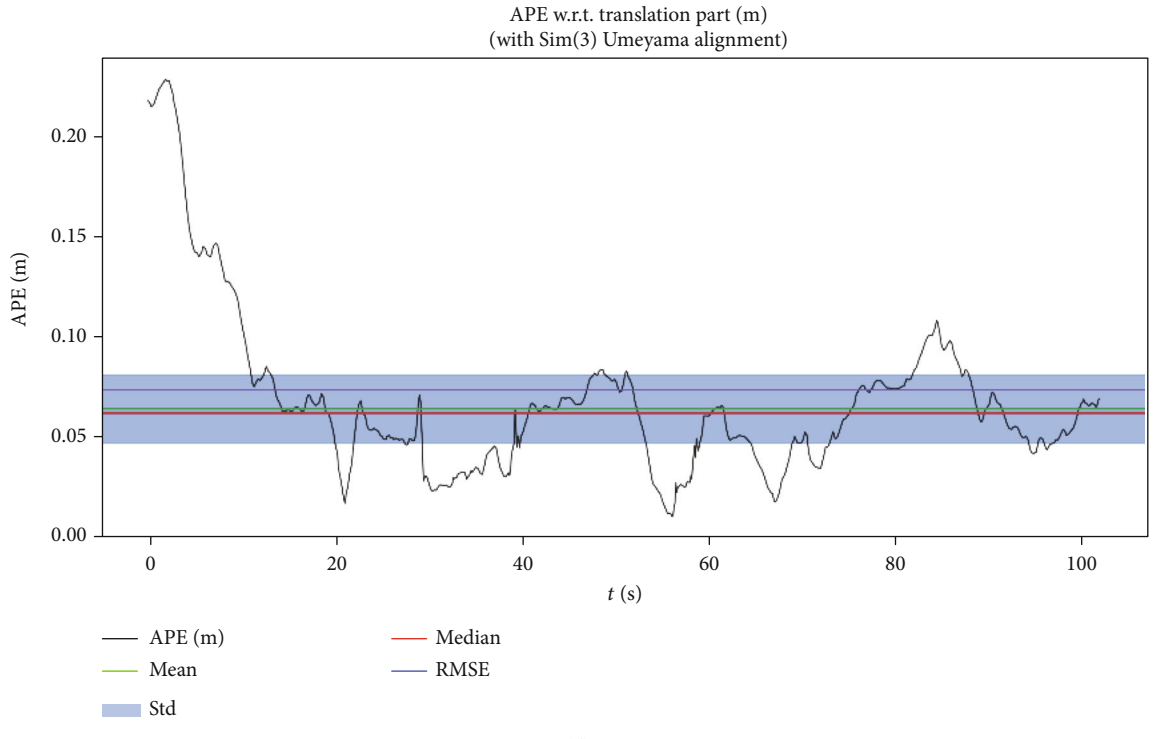


(b)

Figure 8: Continued.

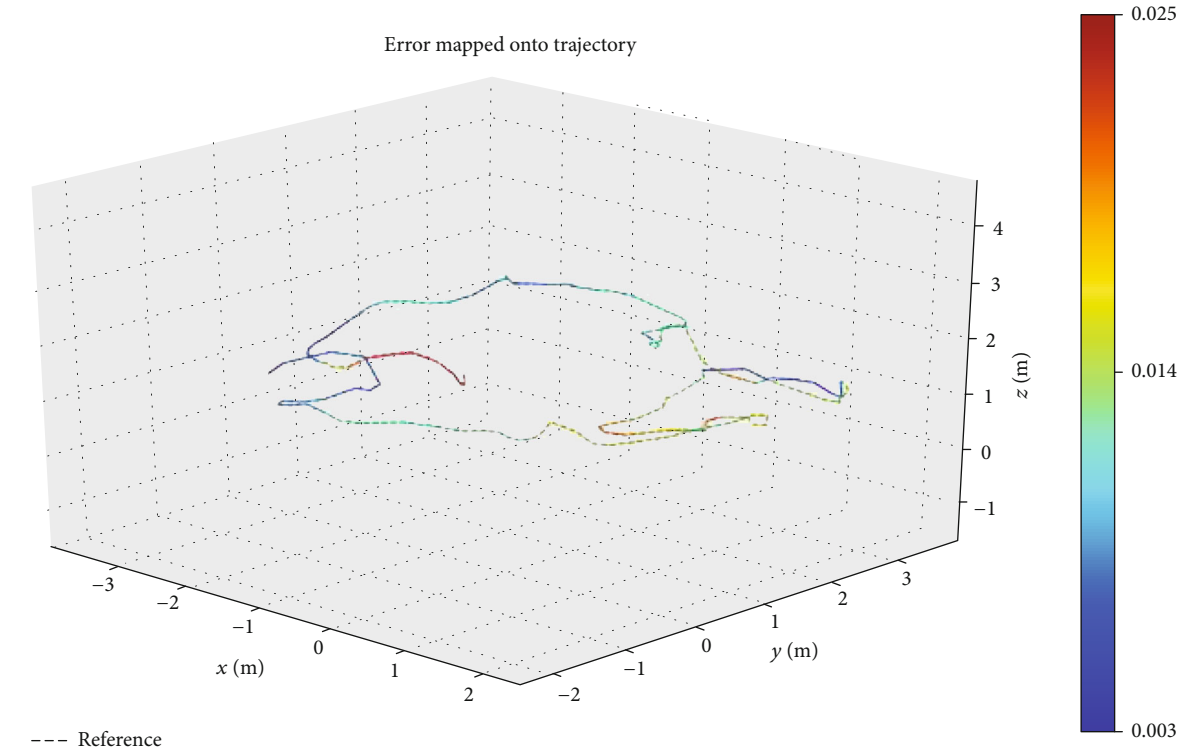


(c)

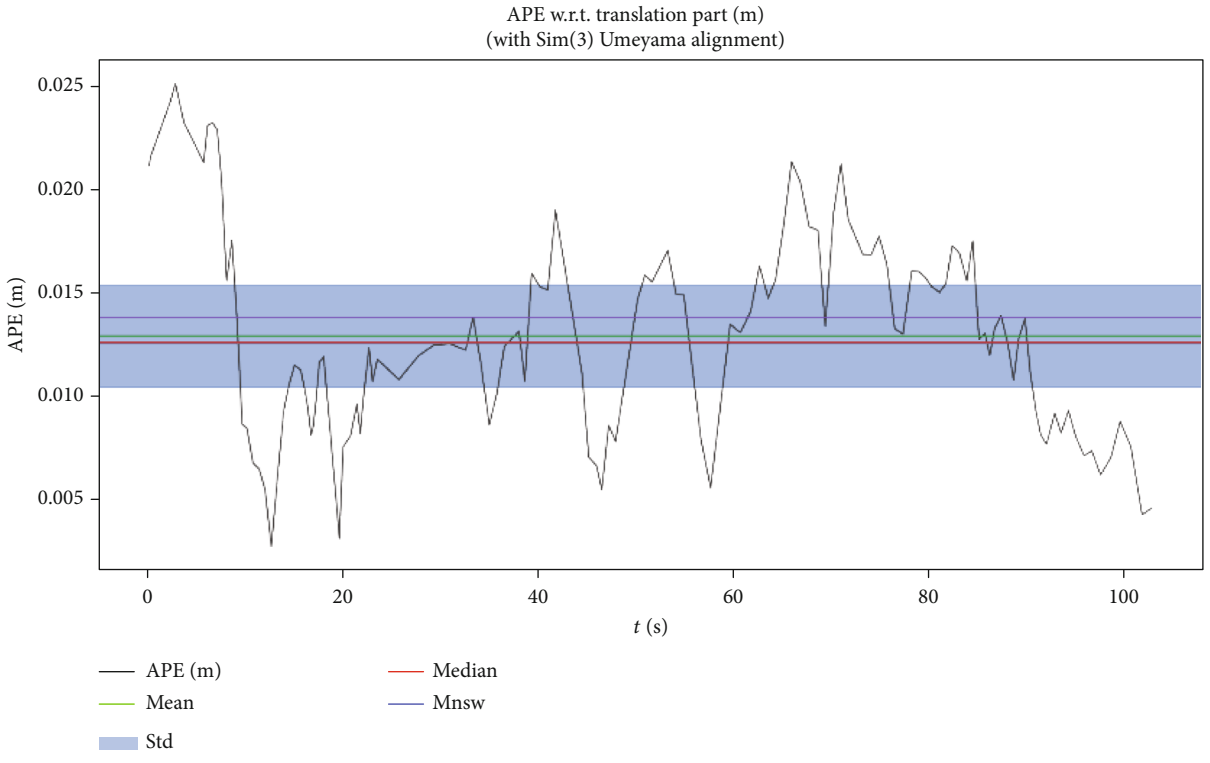


(d)

Figure 8: Continued.

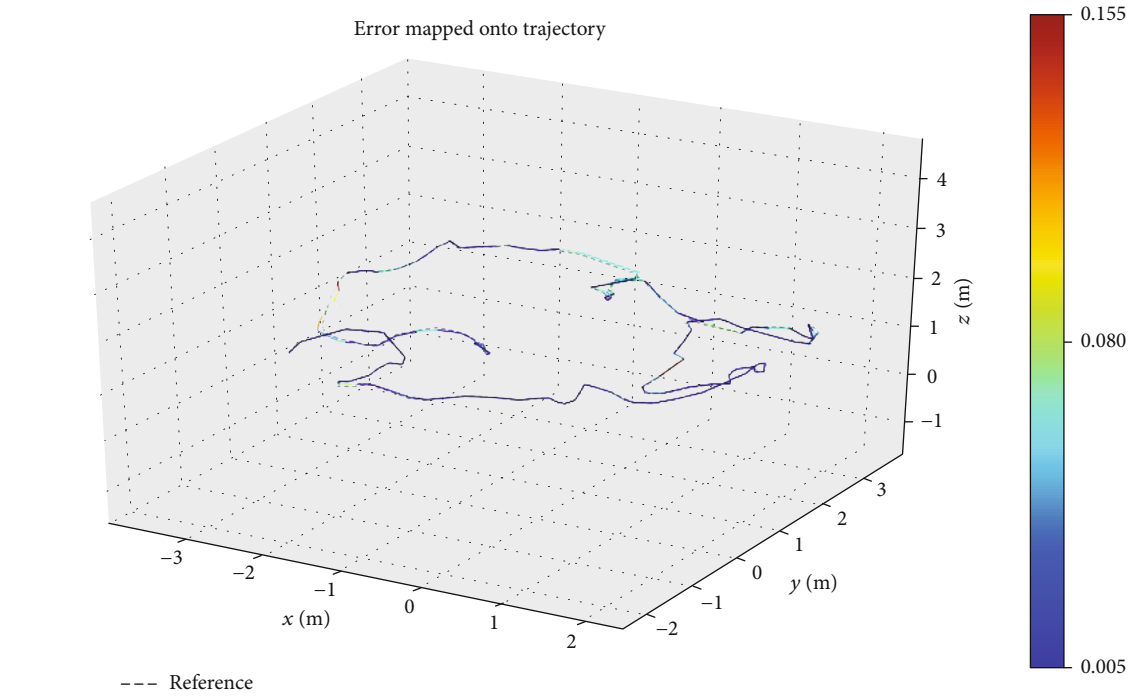


(e)

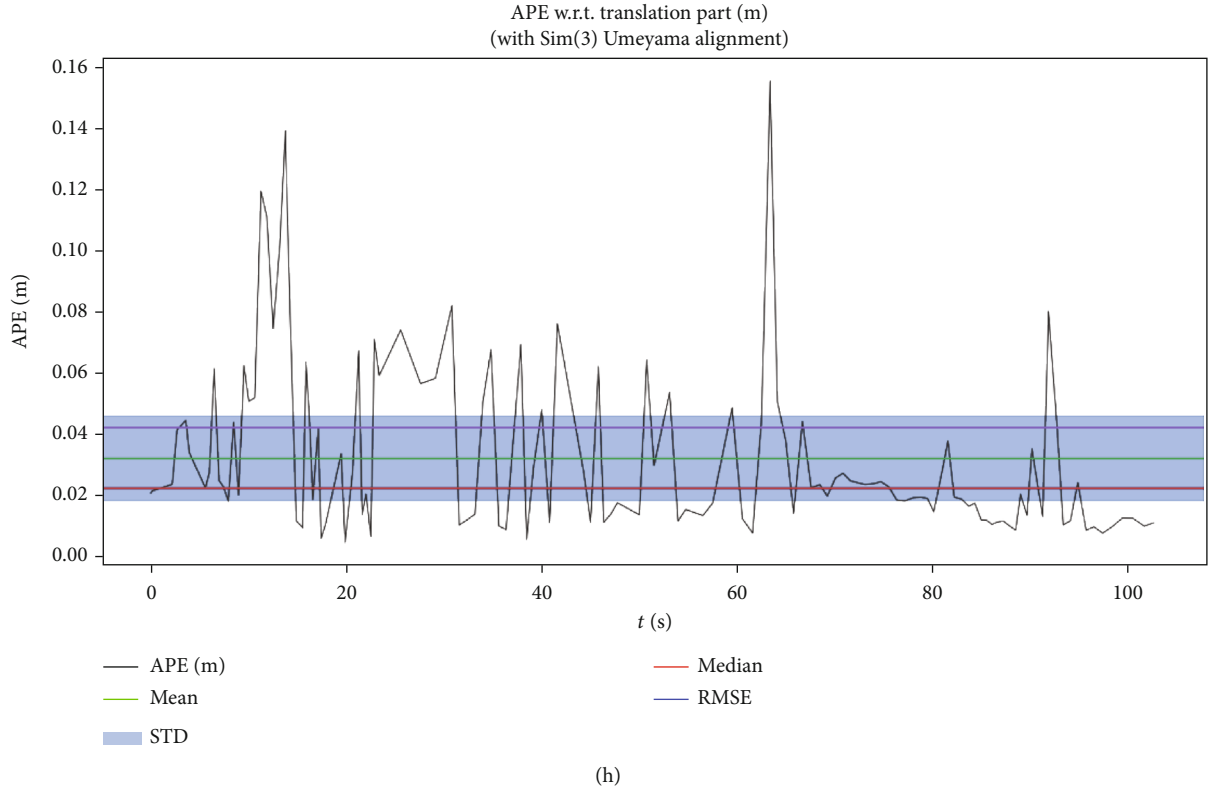


(f)

Figure 8: Continued.



(g)



(h)

Figure 8: (a) The APE results with respect to the translation part of VI-ORBSLAM. (b) The error mapped onto the trajectory of VI-ORBSLAM. (c) The APE results with respect to the translation part of VINS-MONO. (d) The error mapped onto the trajectory of VINS-MONO. (e) The APE results with respect to the translation part of ours. (f) The error mapped onto the trajectory of ours. (g) The APE results with respect to the translation part of ORB-SLAM3. (h) The error mapped onto the trajectory of ORB-SLAM3.

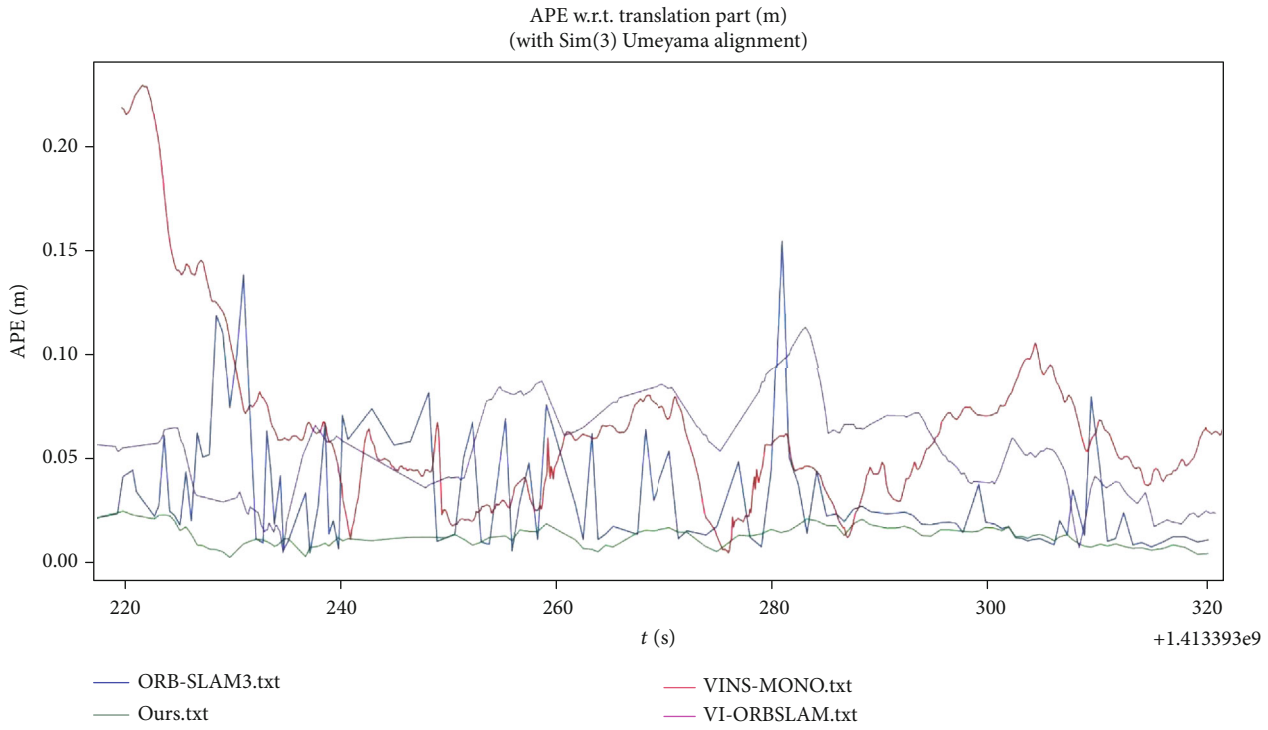


Figure 9: The comparison curve of APE results with respect to the translation part (m). The curves of violet, red, green, and blue represent VI-ORBSLAM, VINS-MONO, Ours, and ORB-SLAM3, respectively.

Table 2: The quantitative comparison results on *V2_01_easy* sequence.

	VI-ORBSLAM	VINS-MONO	Ours	ORB-SLAM3 (mono)
Max	0.113469	0.229729	0.025111	0.155221
Mean	0.051731	0.060444	0.012993	0.035622
Median	0.053289	0.058242	0.012664	0.021165
Min	0.005098	0.005024	0.002810	0.008923
RMSE	0.056971	0.07014	0.013892	0.042956
SSE	0.574481	3.719183	0.024702	1.029630
Std	0.023866	0.035582	0.004915	0.032003

So it can be verified that the improved iterative strategy also has the positive effect on the velocity estimation. Besides, it should be noted that the obtained results are the median of 10 tests on V2_01 sequence. Although we keep the same parameters and the same play speed of dataset (×1.5 speed of the bag file) in each test, the results are still slightly different. For fair comparison, we choose the median value as the evaluating indicator. The boxplots of the 10 tests are shown in Figure 7, and the median values are represented by a horizontal line.

4.4. Localization Accuracy. The accuracy performance of the proposed method is also examined using the whole EuRoC dataset. In this work, the open-source package: evo [34], is adopted to evaluate the VI-SLAM algorithms. The qualitative absolute pose error (APE) results on *V2_01_easy* sequence with respect to the translation are shown in Figures 8(a)–8(h), where the blue colors of (a), (c), (e), and (g) encode the corresponding absolute pose errors of trajectory, and the red color represents the error which is larger than the blue color. Panels (b), (d), (f), and (h) draw the curve of other indicators: mean, median, RMSE, and Std. As shown in Figure 9, the three comparison curves of APE results are also given; it can be seen that our method has the best performance than VI-ORBSLAM and VINS-MONO on APE evaluation. The comparisons of key frame trajectories with ground truth which tested on the *V2_01 sequence* are shown in Figure 10, where the ground truth is represented by a black dotted line, VI-ORBSLAM is represented by a blue line, VINS-MONO is represented by a green line, ORB-SLAM3 is represented by a violet line, and our method is represented by a red line. The quantitative comparison with the values of max, mean, median, min, RMSE, SSE, Std are shown in Table 2.

The quantitative root mean square error (RMSE) of the whole 11 sequences is also the median values with 10

Table 3: Result of the pose estimation in the 11 sequences of the EuRoC dataset. The reported values are the median after 10 executions for each sequence. The italic types indicate the best results. Abbreviations: Trans: translation; Ori: orientation; Ave: average.

Sequence	VI-ORBSLAM		VINS-MONO		OURS		ORB-SLAM3 (mono)	
	Trans. (m)	Ori. (°)	Trans. (m)	Ori. (°)	Trans. (m)	Ori. (°)	Trans. (m)	Ori. (°)
V1_01_easy	*0.027*	*2.112*	0.047	2.733	0.031	2.657	0.043	2.422
V1_02_medium	0.028	1.692	0.066	1.732	0.017	*1.646*	*0.016*	1.673
V1_03_difficult	*x*	*x*	0.180	4.598	*0.021*	*2.512*	0.025	2.632
V2_01_easy	0.053	1.326	0.058	1.525	*0.013*	*1.069*	0.021	1.126
V2_02_medium	0.041	2.580	0.090	2.613	0.030	2.394	*0.015*	*1.180*
V2_03_difficult	0.074	4.012	0.244	5.532	0.049	*3.496*	*0.037*	4.362
MH_01_easy	0.075	2.251	0.084	2.310	*0.031*	*2.036*	0.042	2.155
MH_02_easy	0.084	2.523	0.105	2.753	*0.015*	*2.408*	0.053	3.043
MH_03_medium	0.087	1.353	0.074	1.262	*0.026*	*1.156*	0.043	1.935
MH_04_difficult	0.217	1.306	0.122	1.246	0.110	1.115	*0.099*	*1.106*
MH_05_difficult	0.082	0.559	0.147	*0.395*	*0.056*	0.511	0.071	0.529
Ave.	0.077	1.971	0.111	2.427	*0.036*	*1.909*	0.042	2.015

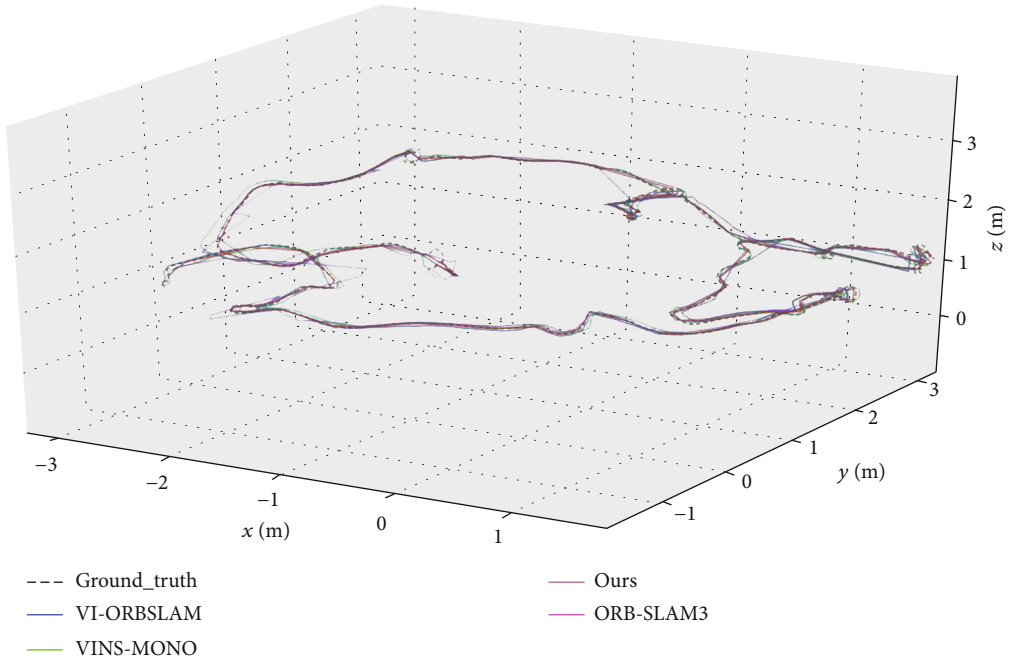


Figure 10: The trajectories of VI-ORBSLAM, VINS-MONO, Ours, and ORB-SLAM3 on *V2_01_easy* sequence. Dotted line: ground truth. Blue line: VI-ORBSLAM. Green line: VINS-MONO. Red line: Ours. Violet line: ORB-SLAM3.

executions in the same computing platform. Our reported results are show in Table 3, and the boxplots are also provided in Figure 11. It should be noted that the estimated pose is usually not in the same coordinate system with the ground truth; we need to give an align process. Practically, we calculate the transformation matrix $S \in \mathrm{Sim}(3)$ from the estimated pose to the ground truth; the APE in ith frame is defined as follows:

$$F_i := Q_i^{-1} S P_i. \tag{17}$$

Then, the absolute translational and orientation RMSE is calculated as follows:

$$\begin{aligned} \mathrm{RMSE}_{\mathrm{trans}}(E_{1:n}, \Delta) &:= \left(\frac{1}{m} \sum_{i=1}^{m} \|\mathrm{trans}(E_i)\|^2 \right)^{1/2}, \\ \mathrm{RMSE}_{\mathrm{ori}}(E_{1:n}, \Delta) &:= \left(\frac{1}{m} \sum_{i=1}^{m} \|\mathrm{ori}(E_i)\|^2 \right)^{1/2}. \end{aligned} \tag{18}$$

From Table 3, it can be known that our method achieves

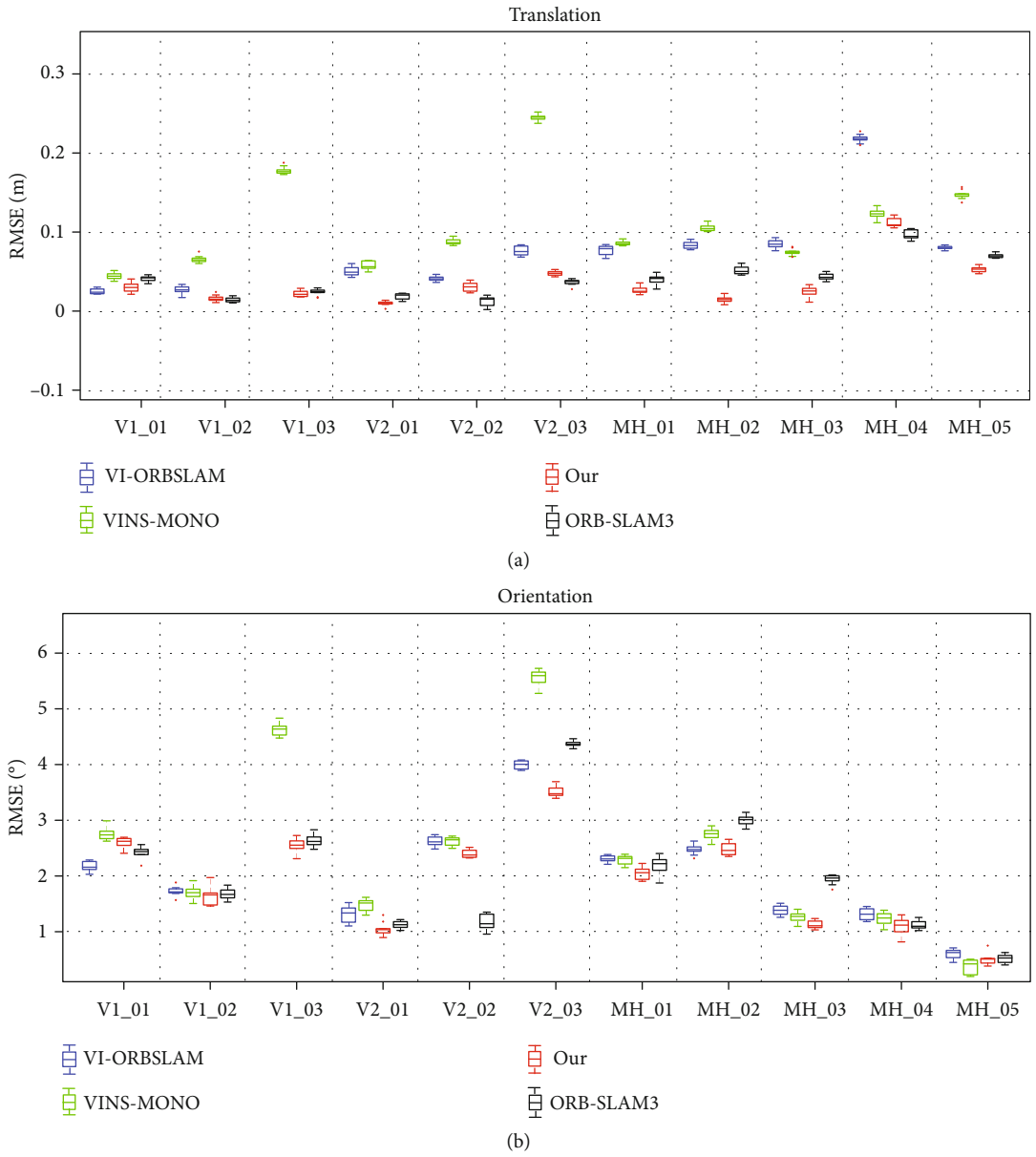


Figure 11: RMSE comparison of pose: (a) translation results; (b) orientation results. The blue box represents VI-ORBSLAM, the green box represents VINS-MONO, the red box represents ours, and the black box represents ORB-SLAM3.

in all sensor configurations more accurate result than the VI-ORBSLAM, VINS-MONO, and ORB-SLAM3. Practically, our method provides the best performance on six sequences for the translation and seven sequences for the orientation. It can be noted that the VI-ORBSLAM could not completely run the *V1_03_difficult* sequence, while our method can run completely. Besides, we list the average values of the all sequences in the last row, where VI-ORBSLAM is calculated as 0.077 m in translation and 1.971 deg in orientation, VINS-MONO is calculated as 0.111 m in translation and 2.427 deg in orientation, ORB-SLAM3 is calculated as 0.042 m in translation and 2.015 deg in orientation, and OURS is calculated with 0.036 m in translation and 1.909 deg in orientation. VI-ORBSLAM and ORB-SLAM3 have a better performance in the testing of several sequences such as *V1_01_easy*, *V1_02_medium*, and *MH_04_difficult*, while our algorithm performs better in other sequences.

5. Conclusions

In this paper, the proposed initialization algorithm is on the basis of the VI-ORBSLAM framework. In order to improve the quality of initialization, an improved trust region-based iterative strategy is proposed. Our method has been verified on the public dataset with faster convergence speed in the initialization stage, as well as the velocity and pose can be estimated more accurately than the original method. At present, the binocular VI-SLAM technology has more practical value in the industrial application. There are several key issues that should be considered, such as the real-time

computing and the online recovery capabilities. In the future works, we will try to apply the proposal for the binocular model, as well as the positioning and mapping of the unmanned vehicles in outdoor environment are being considered.

Data Availability

The datasets used in this article can be found in the following link: https://projects.asl.ethz.ch/datasets/doku.php?id=kmavvisualinertialdatasets.

Conflicts of Interest

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This work was partially supported by the National Natural Science Foundation of China (61673306).

References

[1] S. J. Eagle and S. Stefano, "Visual-inertial navigation, mapping and localization: a scalable real-time causal approach," *The International Journal of Robotics Research*, vol. 30, no. 4, pp. 407–430, 2011.

[2] H. Zheng and G. Huang, "Robocentric visual-inertial odometry," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 6319–6326, Madrid, Spain, 2018.

[3] Y. Lin, F. Gao, T. Qin et al., "Autonomous aerial navigation using monocular visual-inertial fusion," *Journal of Field Robotics*, vol. 35, no. 1, pp. 23–51, 2018.

[4] M. Bloesch, S. Omari, M. Hutter, and R. Siegwart, "Robust visual inertial odometry using a direct EKF-based approach," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 298–304, Hamburg, Germany, 2015.

[5] O. Taragay, S. Supun, and K. Rakesh, "Multi-sensor navigation algorithm using monocular camera, IMU and GPS for large scale augmented reality," in *2012 IEEE international symposium on mixed and augmented reality (ISMAR)*, pp. 71–80, Atlanta, GA, USA, 2012.

[6] P. Li, T. Qin, B. Hu, F. Zhu, and S. Shen, "Monocular visual-inertial state estimation for mobile augmented reality," in *IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pp. 11–21, Nantes, France, 2017.

[7] J. Dong, X. Fei, and S. Soatto, "Visual-inertial-semantic scene representation for 3d object detection," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3567–3577, Hawaii, 2017.

[8] Z. Yang, F. Gao, and S. Shen, "Real-time monocular dense mapping on aerial robots using visual-inertial fusion," in *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4552–4559, Singapore, 2017.

[9] R. Murartal, J. M. Montiel, and J. D. Tardos, "ORB-SLAM: a versatile and accurate monocular SLAM system," *IEEE Transactions on Robotics*, vol. 31, no. 5, pp. 1147–1163, 2015.

[10] R. Murartal and J. D. Tardos, "ORB-SLAM2: an open-source SLAM system for monocular, stereo, and RGB-D cameras," *IEEE Transactions on Robotics*, vol. 33, no. 5, pp. 1255–1262, 2017.

[11] C. Campos, R. Elvira, J. J. Rodríguez, J. M. Montiel, and J. D. Tardós, "ORB-SLAM3: an accurate open-source library for visual, visual-inertial and multi-map slam," 2020, https://arxiv.org/abs/2007.11898.

[12] S. Leutenegger, S. Lynen, M. Bosse, R. Siegwart, and P. Furgale, "Keyframe-based visual-inertial odometry using nonlinear optimization," *The International Journal of Robotics Research*, vol. 34, no. 3, pp. 314–334, 2015.

[13] T. Qin, P. Li, and S. Shen, "VINS-mono: a robust and versatile monocular visual-inertial state estimator," *IEEE Transactions on Robotics*, vol. 34, no. 4, pp. 1004–1020, 2018.

[14] T. Qin, J. Pan, and S. Cao, "A general optimization-based framework for local odometry estimation with multiple sensors," 2019, https://arxiv.org/abs/1901.03638.

[15] L. Von Stumberg, V. Usenko, and D. Cremers, "Direct sparse visual-inertial odometry using dynamic marginalization," in *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2510–2517, Brisbane, QLD, Australia, 2018.

[16] R. Murartal and J. D. Tardos, "Visual-inertial monocular SLAM with map reuse," *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 796–803, 2017.

[17] A. Martinelli, "Closed-form solution of visual-inertial structure from motion," *International Journal of Computer Vision*, vol. 106, no. 2, pp. 138–152, 2014.

[18] J. Kaiser, A. Martinelli, F. Fontana, and D. Scaramuzza, "Simultaneous state initialization and gyroscope bias calibration in visual inertial aided navigation," *IEEE Robotics and Automation Letters*, vol. 2, no. 1, pp. 18–25, 2016.

[19] C. Campos, J. M. M. Montiel, and J. D. Tardos, "Fast and robust initialization for visual-inertial SLAM," in *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1288–1294, Montreal, QC, Canada, 2019.

[20] T. Qin and S. Shen, "Robust initialization of monocular visual-inertial estimation on aerial robots," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4225–4232, Vancouver, BC, Canada, 2017.

[21] Z. Yang and S. Shen, "Monocular visual-inertial state estimation with online initialization and camera-IMU extrinsic calibration," *IEEE Transactions on Automation Science and Engineering*, vol. 14, no. 1, pp. 39–51, 2017.

[22] K. Madsen, H. B. Nielsen, and O. Tingleff, *Methods for non-linear least squares problems*, Informatics and Mathematical Modeling, 2nd edition, 2004.

[23] R. Kümmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard, "g2o: a general framework for graph optimization," in *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3607–3613, Shanghai, China, 2011.

[24] M. Kaess, H. Johnnsson, R. Roberts, V. Leonard, and F. Dellaert, "isam2: Incremental smoothing and mapping using the Bayes tree," *International Journal of Robotic Research (IJRR)*, vol. 31, no. 2, pp. 217–236, 2012.

[25] C. J. Wang, "Dogleg paths and trust region methods with back tracking technique for unconstrained optimization," *Applied Mathematics & Computation*, vol. 177, no. 1, pp. 159–169, 2006.

[26] K. Levenberg, "A method for the solution of certain non-linear problems in least squares," *Quarterly of Applied Mathematics*, vol. 2, no. 2, pp. 164–168, 1944.

[27] D. Marquardt, "An algorithm for least squares estimation on nonlinear parameters. SIAM," *Journal of Applied Mathematics*, vol. 11, 441 pages, 1963.

[28] Z. Zhang, G. Gallego, and D. Scaramuzza, "On the comparison of gauge freedom handling in optimization-based visual-inertial state estimation," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 2710–2717, 2018.

[29] M. Burri, J. Nikolic, P. Gohl et al., "The EuRoC micro aerial vehicle datasets," *The International Journal of Robotics Research*, vol. 35, no. 10, pp. 1157–1163, 2016.

[30] T. Lupton and S. Sukkarieh, "Visual-inertial-aided navigation for high-dynamic motion in built environments without initial conditions," *IEEE Transactions on Robotics*, vol. 28, no. 1, pp. 61–76, 2012.

[31] C. Forster, L. Carlone, F. Dellaert, and D. Scaramuzza, "IMU preintegration on manifold for efficient visual-inertial maximum-a-posteriori estimation," *Robotics Science and Systems*, pp. 1–20, 2015.

[32] R. T. Shahir and H. D. Taghirad, "An improved optimization method for iSAM2," in *Second RSI/ISM Internat-ional Conference on Robotics and Mechatronics (ICRoM)*, pp. 582–587, Tehran, Iran, 2014.

[33] W. Huang, H. Liu, and W. Wan, "An online initialization and self-calibration method for stereo visual-inertial odometry," *IEEE Transactions on Robotics*, vol. 36, no. 4, pp. 1–18, 2020.

[34] M. Grupp, *evo: Python Package for the Evaluation of Odometry and SLAM*, 2017, http://github.com/MichaelGrupp/evo.