remote sensing

MDPI

Article

RNGC-VIWO: Robust Neural Gyroscope Calibration Aided Visual-Inertial-Wheel Odometry for Autonomous Vehicle

Meixia Zhi [1], Chen Deng [2], Hongjuan Zhang [1,3], Hongqiong Tang [4], Jiao Wu [1] and Bijun Li [1,3,*]

[1] State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing, Wuhan University, Wuhan 430079, China; meixiazhi@whu.edu.cn (M.Z.); hongjuanzhang@whu.edu.cn (H.Z.); wujiaors@whu.edu.cn (J.W.)
[2] State Key Laboratory of Geo-Information Engineering, Xi'an Research Institute of Surveying and Mapping, Xi'an 710054, China; dr_dengchen@163.com
[3] Engineering Research Center for Spatio-Temporal Data Smart Acquisition and Application, Wuhan University, Wuhan 430079, China
[4] The Department of Navigation Engineering, Naval University of Engineering, Wuhan 430079, China; hqtang@foxmail.com
* Correspondence: lee@whu.edu.cn

Abstract: Accurate and robust localization using multi-modal sensors is crucial for autonomous driving applications. Although wheel encoder measurements can provide additional velocity information for visual-inertial odometry (VIO), the existing visual-inertial-wheel odometry (VIWO) still cannot avoid long-term drift caused by the low-precision attitude acquired by the gyroscope of a low-cost inertial measurement unit (IMU), especially in visually restricted scenes where the visual information cannot accurately correct for the IMU bias. In this work, leveraging the powerful data processing capability of deep learning, we propose a novel tightly coupled monocular visual-inertial-wheel odometry with neural gyroscope calibration (NGC) to obtain accurate, robust, and long-term localization for autonomous vehicles. First, to cure the drift of the gyroscope, we design a robust neural gyroscope calibration network for low-cost IMU gyroscope measurements (called NGC-Net). Following a carefully deduced mathematical calibration model, NGC-Net leverages the temporal convolutional network to extract different scale features from raw IMU measurements in the past and regress the gyroscope corrections to output the de-noised gyroscope. A series of experiments on public datasets show that our NGC-Net has better performance on gyroscope de-noising than learning methods and competes with state-of-the-art VIO methods. Moreover, based on the more accurate de-noised gyroscope, an effective strategy for combining the advantages of VIWO and NGC-Net outputs is proposed in a tightly coupled framework, which significantly improves the accuracy of the state-of-the-art VIO/VIWO methods. In long-term and large-scale urban environments, our RNGC-VIWO tracking system performs robustly, and experimental results demonstrate the superiority of our method in terms of robustness and accuracy.

Keywords: gyroscope calibration; yaw attitude correction; deep learning; multi-sensor fusion; vehicle localization; visual-inertial-wheel odometry

Citation: Zhi, M.; Deng, C.; Zhang, H.; Tang, H.; Wu, J.; Li, B. RNGC-VIWO: Robust Neural Gyroscope Calibration Aided Visual-Inertial-Wheel Odometry for Autonomous Vehicle. *Remote Sens.* **2023**, *15*, 4292. https://doi.org/10.3390/rs15174292

Academic Editors: Mennatullah Siam and Xinshuo Weng

Received: 27 July 2023
Revised: 25 August 2023
Accepted: 28 August 2023
Published: 31 August 2023

1. Introduction

Using low-cost multi-sensors for high accuracy and robust positioning is a challenging task for ground vehicles in GPS-denied urban environments [1]. Approaches fusing visual, inertial, and wheel encoder measurements called visual-inertial-wheel odometry (VIWO) have received a lot of attention in recent years. Compared with traditional visual odometry (VO) [2–6] and visual-inertial odometry (VIO) [7–14], the additional wheel measurements can provide true-scale velocities to render the scale of VO/VIO observable, especially when the vehicle is in plane motion with constant acceleration, which is the common motion

situation of the ground vehicle [15]. Therefore, integrating the wheel measurements with VIO can further improve the accuracy and robustness of vehicle localization.

However, similar to VIO, VIWO still cannot avoid the pose drift caused by the low-precision attitude acquired by the gyroscope of a low-cost IMU in complex environments. When VIO or VIWO experiences visual degradation scenes, such as weak or repetitive textures, shimmering lighting, or dynamic change, the visual information cannot accurately correct for the IMU bias [16]. Limited by the accuracy of an IMU based on a micro-electro-mechanical system (MEMS), the attitude integrated from the original gyroscope measurements suffers from large drift for a long time, especially due to the bias instability (BI) and angular random walk (ARW) resulting from the noise and thermal effect [17]. The errors in attitude will further lead to the rapid accumulation of positional errors solved by systems that conduct motion estimation through past state recursion, such as an inertial navigation system (INS), VIO, VIWO, etc. In particular, for the INS, the positional errors of the system grow cubically in time caused by the gyroscope bias [18]. In VIO/VIWO systems [7,19–21], inaccurate positions will be added to the joint optimization framework as important residual constraints. Even though the existing methods have applied the pre-integration technique, it will inevitably decrease the pose accuracy calculated by the solver and further lead to long-term pose drift.

Therefore, the accuracy of IMU measurements, especially the quality of gyroscope data, is one of the important factors affecting the overall orientation and position accuracy of the VIO/VIWO system over a longer period. Thus, correcting and compensating for the errors of the MEMS IMU gyroscope is crucial to obtain long-term attitude stability, which further helps to improve the overall accuracy of the VIO/VIWO system. Currently, for VIO/VIWO systems, to limit the long-term drift as much as possible, one approach is to introduce the global navigation satellite system (GNSS) [13,22,23]. The GNSS can provide global positioning information, helping to eliminate cumulative errors. Nevertheless, satellite signals are susceptible to multipath fading and shadow effects, which affect the accuracy of the GNSS, especially in the city canyon, tunnel, or underground parking lot environments. Another method is to employ loop closing, which can effectively reduce the positioning drift but is hard to apply in large-scale outdoor environments [24,25]. Among the commonly proposed methods, some other researchers also employ a pre-built map to bound the long-term errors by matching the map features with the on-the-fly sensor readings [26]. In contrast to these works, de-noising the gyroscope measurements to improve long-term positioning accuracy does not need to rely on the additional sensors, specific motion patterns, and prior maps.

In recent years, deep-learning-based methods have been introduced into inertial navigation systems, where motion can be inferred from the IMU, such as pedestrian motion state estimation [27,28]. These studies show that human motion priors can be learned from IMU measurements through networks. Meanwhile, the results show that the accuracy based on deep learning is comparable to that of the VIO algorithm. At present, inertial learning is rarely used in robots. Reference [29] uses a CNN to correct IMU noise and bias and directly integrates the de-noised IMU to obtain the orientation. The de-noised gyroscope is also applied to the VIO methods. Similar efforts directly replace the original measurements in the VIO approach with network outputs [30–32]. The way of fusing network outputs with VIO is very simple and cannot significantly improve the accuracy of the traditional VIO methods.

In this paper, aiming at reducing the long-term drift of VIWO deployed for autonomous vehicles, we present a tight-coupled nonlinear optimization method, which effectively integrates the robust neural gyroscope data with the traditional visual-inertial-wheel odometry to perform long-term pose estimation in complex driving environments. Our approach is motivated by the observation that calibrating the errors of MEMS gyroscope measurements based on neural networks can provide more accurate attitude estimation, which can further provide better directional constraints for the VIWO method to cure long-term drift. The main contributions of this work include:

(1) A robust neural calibration network for low-cost IMU gyroscopes called NGC-Net is proposed, which leverages the temporal convolutional network to extract the error features from the raw IMU. By effective data enhancement strategy, well-designed network structure, and multiple losses considered, our experiments show the proposed NGC-Net can achieve better de-noising performance.
(2) We design an effective fusion strategy to combine the advantages of network outputs and VIWO methods and further propose a novel multi-sensor fusion tracking method to reduce the long-term drift using the heading obtained by our NGC-Net outputs.
(3) Through a series of experiments on public datasets, our NGC-Net has better performance than both learning methods and competes with VIO methods. We implement the RNGC-VIWO system and validate the proposed method in complex urban driving datasets. Compared with state-of-the-art methods, our method can significantly improve the accuracy and robustness of vehicle localization in long-term and large-scale areas.

The structure of this paper is as follows. Section 1 is the introduction. Section 2 presents the related work. Section 3 elaborates on the details of our method. The experimental results are presented in Section 4. Finally, a brief conclusion and outlook are given in Section 5.

2. Related Work

Over the past decades, localization methods by fusing multi-model sensors have become popular research. Autonomous driving system (ADS) vehicles equipped with multi-modal sensors (such as cameras, GNSS, LiDAR) provide multi-source sensor data to perceive the surrounding traffic environment and obtain vehicle position, speed, and attitude information [33]. We mainly review related work on vision-based methods and IMU correction approaches.

2.1. Vision-Aided or -Based Methods

Over the past few decades, great progress has been made in the research of monocular visual odometry/SLAM. The typical studies include PTAM [2], SVO [3], and ORB-SLAM [5]. While visual odometry can achieve high precision in ideal environments, it often fails when dealing with untextured areas, motion blur, and severe lighting variations. In addition, it cannot estimate the true scale of the scene, resulting in scale drift.

To reduce the dependence on visual information and obtain high-precision and robust localization, researchers have proposed visual-inertial odometry (VIO) that combines visual information with IMU measurements. According to the VIO algorithm framework, it can be divided into filter-based and optimization-based approaches. Popular filtering methods include MSCKF [34] and Open-VINS [7], which achieve great estimation performance. The typical optimization method is OKVIS [8], which infers a probabilistic cost function that minimizes visual reprojection errors and pre-integrated IMU errors. VINS-Mono [11] is another robust visual-inertial SLAM system. Compared to OKVIS, it has robust initialization, relocalization, and pose graph optimization. VINS-Fusion [12,13] is an extension of VINS-Mono that supports multiple sensor combinations, including fusing VINS with GPS. Under the constraints of IMU pre-integration, ORB-SLAM3 optimizes camera poses in a covisibility graph [14]. Although these VIO systems achieve impressive accuracy and robustness, they will suffer from large drift if running in complex environments for a long time, which is insufficient for autonomous vehicle applications [35].

Recent research has incorporated wheel encoder measurements into VIO to improve positioning accuracy and robustness. Reference [15] demonstrates that the VIO system could not estimate the correct scale with constant acceleration motion, nor could it estimate good roll and pitch without rotational motion. To address these issues, wheel encoder readings with the VIO are integrated in a tightly coupled framework to make the scale observable. Reference [19] proposes a multi-sensor fusion SLAM algorithm using monocular vision, IMU, and wheel encoder measurements. Reference [20] presents a tightly coupled

monocular visual SLAM using wheels and a MEMS gyroscope and introduces the wheel odometer error term into the optimization process, which is a complete SLAM framework and can increase the accuracy and robustness of localization. However, the use of wheel odometers for positioning is mainly focused on "planar" applications, which are usually only suitable for indoor environments. In [21], IMU-odometer pre-integration is introduced into the initialization and optimization of VIO systems, and an online extrinsic calibration is designed to improve the accuracy dramatically. Reference [36] propagates the system state by angular from a gyroscope and linear velocity obtained from a wheel odometer and also adds GPS directly to make position observable. In [37], an effective MSCKF-based VIWO method is developed to fuse IMU, camera, and pre-integrated wheel measurements, which calibrates the intrinsic and spatiotemporal extrinsic parameters between sensors online to further improve the overall accuracy of the system. Although incorporating wheel encoder measurements can improve the accuracy of the VIO system, it still cannot avoid long-term drift caused by the limitations of vision and the low accuracy of MEMS IMUs in complex driving environments.

2.2. IMU Correction Methods

The IMU measurement output is often modeled as a linear polynomial equation for systematic errors, such as constant bias, scale factor, and axis misalignment error. The coefficient parameters are determined using high-precision external equipment, such as a three-axis turntable. Reference [38] presents a calibration approach for low-precision MEMS IMUs using a nonlinear model and the transformed unscented Kalman filter (TUKF) with a turntable. Reference [39] proposes a self-calibrated visual-inertial odometry to estimate the IMU scale factor and axis misalignment error using an extended Kalman filter-based pose estimator. In addition, the vehicle chassis sensors and vehicle kinematics/dynamics reflect the vehicle's own characteristic information such as wheel speed, steering wheel angle, and sideslip angle estimation, which provide the longitudinal and lateral vehicle velocities to correct IMU errors [40–42]. For example, according to the error dynamics and observation equations, the degree of the observability of the yaw misalignment is analyzed, and the yaw misalignment of the IMU is estimated by using a Kalman filter [43].

Many researchers have introduced deep learning techniques into the visual-inertial navigation field to improve navigation and positioning accuracy. VI-Net uses the LSTM network to extract motion features from the raw IMU data and fuses these features directly with image features for pose estimation [44]. IONet uses a two-layer LSTM network to learn the IMU measurements of smartphones and to track user movement over time [28]. A tight learned inertial odometry (TLIO) is proposed to extract original IMU features using ResNet and estimate 3D displacement and its uncertainty, allowing them to be fused tightly in an extended Kalman filter (EKF) to estimate pose, speed, and biases for pedestrian dead reckoning [27]. RNIN-VIO also uses an LSTM-style IMU neural network to learn pedestrian movement priors from raw IMU data and fuses network outputs and visual-inertial information into the EKF to improve the robustness of VIO [45]. To sum up, these position estimation algorithms based on deep learning achieve good results compared with traditional methods in different scenarios. With the increasing complexity of scenes, accurate position estimation alone can no longer meet the existing task requirements. Recently, the use of deep learning technology to calibrate IMU errors has begun to attract the attention of researchers, and existing studies have shown that MEMS IMU calibration based on deep learning is feasible. Reference [46] proposes a convolutional neural network (CNN) to reduce accelerometer error. Reference [47] uses a neural structure based on an LSTM to estimate the attitude and introduces an efficient algorithm to calibrate the bias of the gyroscope. In [30], the depthwise separable convolution is used to compensate for IMU errors, which can improve the localization accuracy of inertial navigation. Reference [32] uses dilation convolution for raw IMU feature extraction and outputs the gyroscope error compensation. These network estimation results in gyroscopes are also applied to VIO

methods [29–32], but the fused accuracy gains over traditional VIO are very limited, which does not fully reflect the advantages of the network calibration and VIO methods.

3. Method

3.1. Overview

3.1.1. System Overview

Based on VINS-Mono [11], the proposed approach applies a new IMU learning network and fusing strategy to achieve more efficient and accurate pose estimation. The framework is shown in Figure 1. Unlike previous methods [29–32], the de-noised gyroscope outputs are simply directly input into the mature process of the existing VIO method. Our method effectively fuses the de-noised gyroscope outputs with the existing VIO method in different modules, better combining the advantages of deep learning with traditional VIO methods. Our system consists of four modules: measurements preprocessing, initialization, nonlinear optimization, and yaw attitude correction.

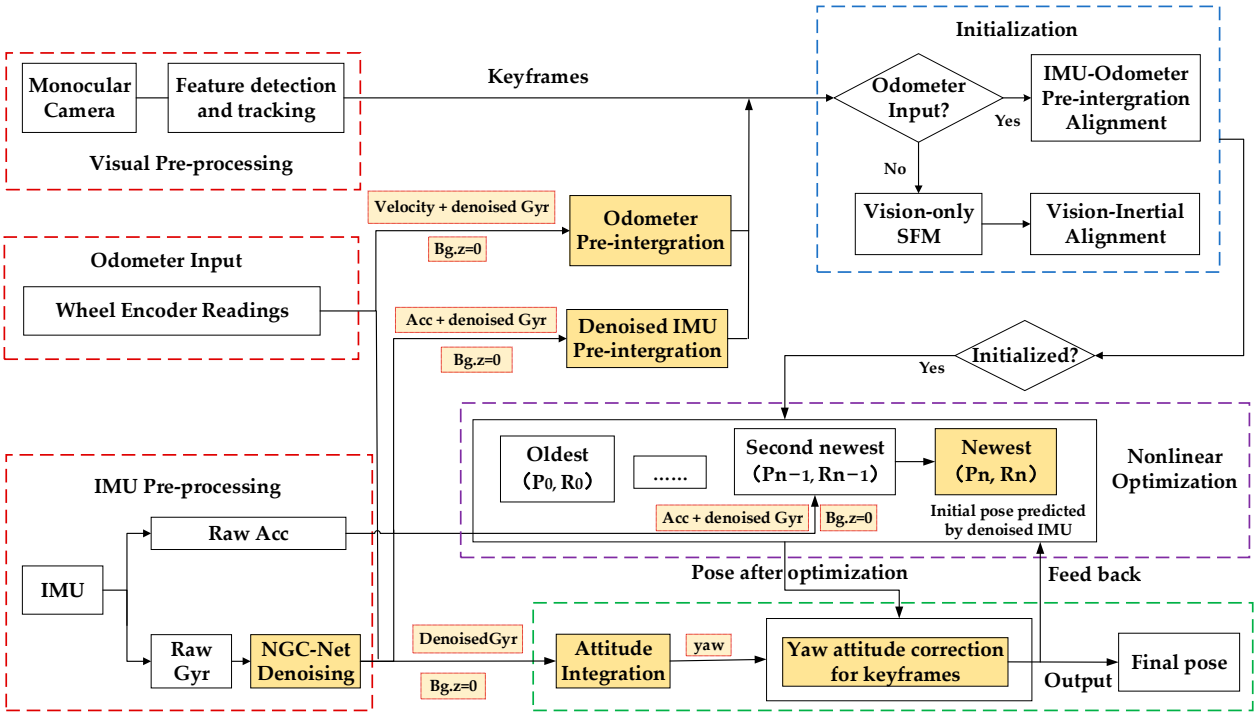


Figure 1. The overall framework of our proposed RNGC-VIWO method. The highlighted modules demonstrate the special contributions of our work.

In the data preprocessing phase, for IMU inputs, a robust neural calibration network for low-cost IMU gyroscope called NGC-Net is first proposed. Then, the IMU pre-integrations between two consecutive frames are calculated using the de-noised gyroscope outputs and raw accelerometer measurements. The wheel odometer pre-integrations between two consecutive frames are also calculated using the de-noised gyroscope outputs and the velocities transformed by wheel encoder readings. To maintain the best orientation accuracy, the z-axis bias of the gyroscope obtained from the nonlinear optimization is not used, which is set to zero.

In the initialization phase, thanks to our NGC-Net, we only solve the gravity direction in the first frame and the optimized velocity for each frame and no longer solve the gyroscope bias, which can help to improve the speed of initialization.

In the nonlinear optimization stage, combining the advantages of VIO and NGC-Net, we do not use the z-axis bias of the gyroscope calibrated online and only update the pre-integration of IMU and odometer according to the calibrated x-axis and y-axis biases of the gyroscope.

In the phase of yaw attitude correction, the horizontal attitude obtained by integrating the de-noised gyroscope angular velocity is used to correct the yaw attitude of each optimized frame, and the corrected poses are further fed back into the sliding window for the following keyframe optimization and output as the final pose.

3.1.2. Notation

We now define notations and coordinate systems throughout the paper. The coordinate systems of the sensors are illustrated in Figure 2. The camera, IMU, and wheel encoder are mounted on a vehicle, and the wheel encoder is installed on the left rear wheel.

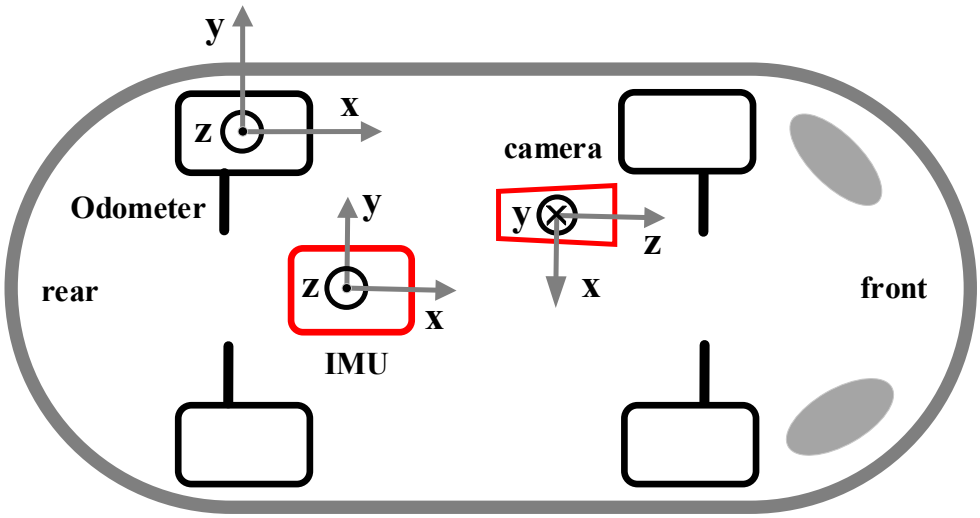


Figure 2. Coordinate systems of sensors mounted on the vehicle.

Figure 2 shows the top view of the coordinate systems, in which $\odot$ denotes the axis perpendicular to the paper plane outward, and $\otimes$ denotes the axis perpendicular to the paper plane inward. Each sensor has its individual local coordinate system, and we use $(\cdot)^W$, $(\cdot)^C$, $(\cdot)^I$, and $(\cdot)^O$ to denote the world coordinate system, the camera coordinate system, the IMU coordinate system, and the odometer coordinate system, respectively. The world coordinate system $(\cdot)^W$ is the coordinate system of the first IMU frame after gravity rotation correction when the system is initialized, and it is fixed once initialized. $G^W = [0; 0; g]^T$ is the prior gravity vector in the world coordinate system, and here $g = 9.81007$. We use $(\cdot)^{C_k}$, $(\cdot)^{I_k}$, and $(\cdot)^{O_k}$ to represent measurements or estimations in the camera frame, IMU frame, and odometer frame corresponding to the time of the k^{th} image frame, respectively. Finally, we denote $(\hat{\cdot})$ as the estimated states or the original measurements of a certain quantity with some noise, and $(\tilde{\cdot})$ as the de-noised measurements of a certain quantity.

3.2. Gyroscope Error Calibration Based on Deep Learning

3.2.1. Gyroscope Correction Model

A typical low-cost inertial measurement unit (IMU) usually consists of a three-axis gyroscope and a three-axis accelerometer. The gyroscope measures angular velocity $\hat{\omega}_i$, the accelerometer measures acceleration $\hat{a}_i$ in the IMU coordinate system, and the output model of the IMU sensor can be represented as seen below [48,49]:

$$\begin{bmatrix} \hat{\omega}_i \\ \hat{a}_i \end{bmatrix} = C \begin{bmatrix} \omega_i \\ a_i \end{bmatrix} + \begin{bmatrix} b_{\omega_i} \\ b_{a_i} \end{bmatrix} + \begin{bmatrix} n_{\omega_i} \\ n_{a_i} \end{bmatrix} \tag{1}$$

where ω_i and a_i are the actual values of the gyroscope angular velocity and accelerometer linear acceleration, which are affected by the gyroscope bias b_{ω_i}, the accelerometer bias b_{a_i}, and the corresponding additive noises n_{ω_i} and n_{a_i}. The additive noises are zero-mean

white Gaussian noises. C is the intrinsic calibration matrix ($C \approx I_6$) for the IMU model, which can be given by

$$C = \begin{bmatrix} L_\omega M_\omega & C_* \\ 0_{3\times3} & L_a M_a \end{bmatrix} \approx I_6 \tag{2}$$

where L_ω and M_ω and L_a and M_a represent the scale factor and axis misalignment matrix of the gyroscope and the accelerometer, respectively, all of which are approximately equal to the identity matrix. C_* denotes the linear accelerations on the gyroscope measurements, i.e., g-sensitivity [29], and it is approximately equal to $0_{3\times3}$. The gyroscope angular velocity after error calibration can be shown as (3)

$$\widetilde{\omega}_i = C_\omega^{-1}(\hat{\omega}_i - C_* a_i - b_{\omega_i} - n_{\omega_i}) = C_\omega^{-1}\hat{\omega}_i - \delta\omega_i \tag{3}$$

where $\widetilde{\omega}_i$ is the de-noised gyroscope angular velocity, and $\hat{\omega}_i$ is the original angular velocity of the gyroscope. Both C_ω^{-1} and $\delta\omega_i$ affect gyroscope calibration and attitude estimation. C_ω^{-1} contains scale factor and axis misalignment errors of the gyroscope, i.e., $C_\omega^{-1} = (S_\omega M_\omega)^{-1}$. We define $\delta\omega_i = C_\omega^{-1}(b_{\omega_i} + n_{\omega_i} + C_* a_i)$ as the main gyroscope error correction term, which is a time-varying error. In addition, the acceleration a_i can also provide information for the angular velocity correction, since it has a certain influence on the error $\delta\omega_i$ and should be included as part of the input of the neural network to reduce the impact on the error.

We now need to estimate $\delta\omega_i$ and C_ω^{-1}. The neural network described in Section 3.2.3 predicts $\delta\omega_i$ by leveraging the gyroscope and accelerometer measurements in a past local window of size N. In most cases, C_ω^{-1} can be taken as an identity matrix, so we set it as the static parameter initialized at I_3 and then set it as the trainable variable optimized in the training process. Here, the form of network can be expressed as

$$\delta\omega_i = f((\hat{a}_{i-N}, \hat{\omega}_{i-N}), \cdots, (\hat{a}_i, \hat{\omega}_i)) \tag{4}$$

where $f(\cdot)$ represents the nonlinear function defined by the proposed NGC-Net. $\hat{a}_{i-N}$ and $\hat{\omega}_{i-N}$ respectively represent the raw IMU at the time corresponding to the (i-N)$^{\text{th}}$ inertial frame.$\hat{a}_i$ and $\hat{\omega}_i$ are the IMU at the time corresponding to the i^{th} inertial frame, and N is the size of the local window.

After modeling the gyroscope correction using the deep neural network, the NGC-Net parameters are updated by calculating the loss function between the ground truth and the predicted attitude during the training process. By using iterative training, a well-trained network model can be obtained to compute the de-noised gyroscope angular velocity $\widetilde{\omega}_i$ by subtracting $\delta\omega_i$ from $C_\omega^{-1}\hat{\omega}_i$, and further, more accurate attitude can also be obtained using integration of the de-noised gyroscope outputs.

3.2.2. Data Preprocessing

To improve the performance of our network, we also adopt data enhancement strategy, which can get diverse data and help avoid overfitting. Firstly, considering different types of IMUs may have different white Gaussian noise and biases. In order to reduce the sensitivity to different IMU noises in the training phase, Gaussian white noise and biases are added randomly to the raw IMU data. Secondly, as the network has only access to the IMU measurements, it suffers from the same observability problem that yaw is not observable. To deal with this unobservability problem, a yaw angle rotation is randomly added to each sample data to learn the invariance characteristics of yaw. Thirdly, in order to strengthen training, referring to [50], we also integrate 20%, 40%, 60%, 80%, and 100% measurements to get multiple losses. This is because for multi-sensor fusion algorithm, IMU integration may be required in different durations. It is worth noting that only the enhanced data are fed into the network.

3.2.3. Network Structure

Previous work has shown that temporal convolutional network (TCN) convincingly outperforms baseline recurrent architectures to solve time-series data modeling problems. Compared with canonical networks such as LSTMs and GRUs, TCN has the advantages of stable gradient and flexible acceptance field, requiring less computing resources, and its structure is simpler and clearer [51].

As shown in Figure 3, we design the NGC-Net based on the architecture of TCN, which is composed of six residual blocks with 32, 64, 128, 256, 72, and 36 channels, respectively, and an output layer with a 1D convolutional layer. We set the same kernel size k = 5 for each residual block and dilation factors d = 1, 2, 4, 8, 16, 32, which increase exponentially with the depth of the network (i.e., d = O(b^m), b = 2) at level m of the network [52]). The kernel size k and the dilation factors d determine the receptive field N of NGC-Net, which is 505 in Equation (4).

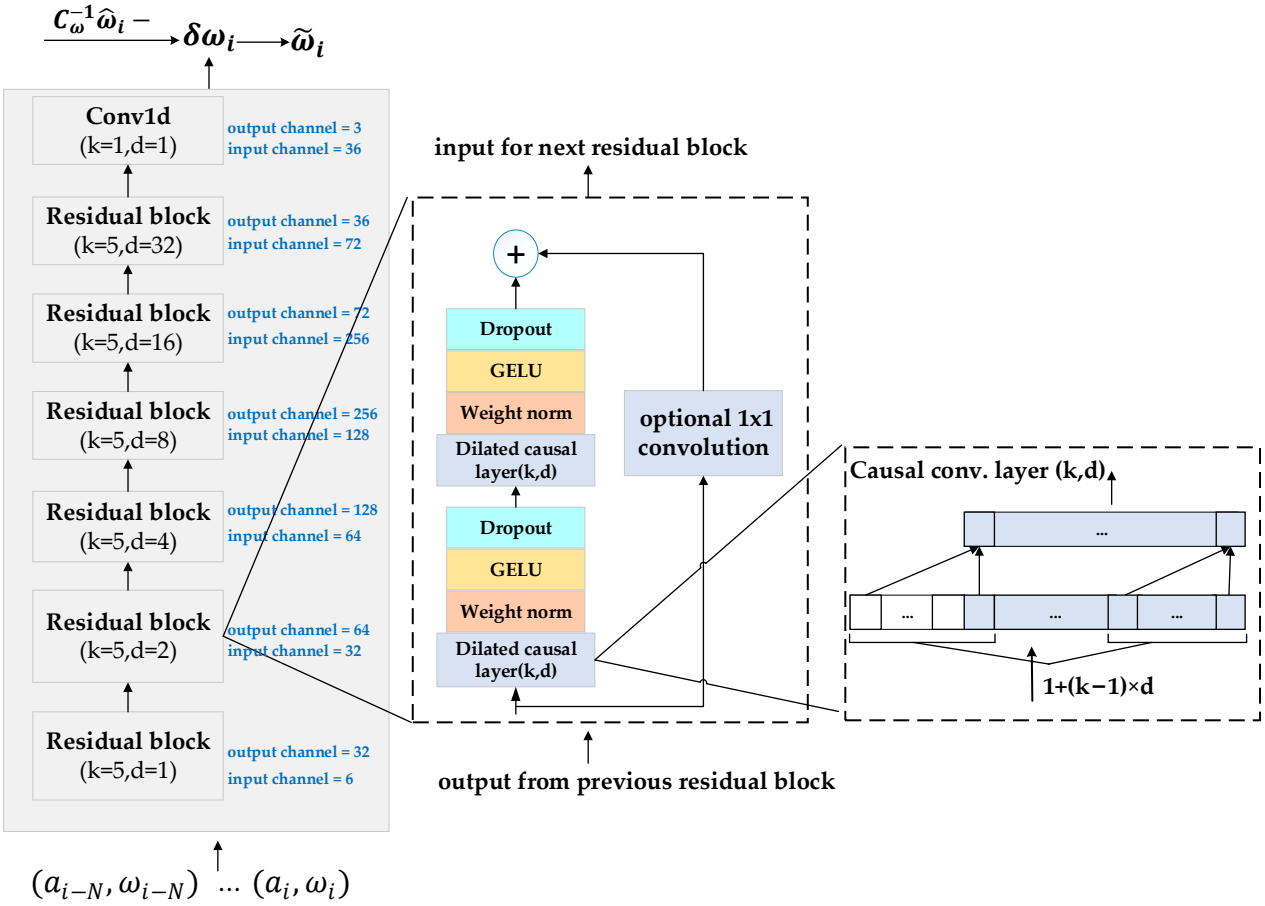


Figure 3. Network structure of the proposed NGC-Net.

As shown in Figure 3, each residual block contains two layers of dilated causal convolution. Gaussian error linear unit (GELU) [53] function is adopted to extract hidden features from the data, and weight normalization is performed for each residual block [54]. In order to avoid overfitting, a spatial dropout [55] is added after each dilated convolution for regularization. Furthermore, a residual connection is introduced into each residual block to maintain the stabilization of deeper and larger TCN [51]. We note that our NGC-Net is relatively preliminary and can be further optimized in the future. However, our results show that the current network is already superior to competing methods, so the improved accuracy of the de-noised gyroscope readings provides a more robust justification for the proposed multi-sensor fusion strategy.

3.2.4. Loss Function

We use the Log-cosh loss for gyroscope correction in the proposed NGC-Net. Theoretically, the loss function should be defined as the error between the real angular velocity and the estimated value. However, on the one hand, the corrected angular velocity is generally consistent with the IMU frequency, in hundreds of Hz, much higher than the frequency of the current best tracking systems, which are generally accurate at 20–120 Hz [29]. On the other hand, supervised learning methods require ground truth references for training, but many datasets usually provide attitude information rather than true gyroscope measurements. Thus, in order to conveniently calculate the loss and achieve better calibration performance, similar to [29], we also use the integrated orientation increment errors rather than the angular velocity errors to construct the loss function. The integral orientation increments can be expressed as (5):

$$\delta\widetilde{R}_{i,i+l} = \widetilde{R}_i^T \widetilde{R}_{i+l} = \prod_{j=i}^{i+l-1} exp\left(\widetilde{\omega}_j \delta t\right) \tag{5}$$

where $exp(\cdot)$ is the exponential map in the $SO(3)$. $\widetilde{\omega}_j$ is the corrected gyroscope angular velocity. δt is the time interval between two consecutive gyroscope measurements. l is the integral increment length.

The Log-cosh loss for a given l can be computed as follows:

$$\mathcal{L}_s = \sum_i Log - cosh\left(log\left(\delta R_{i,i+l} \delta\widetilde{R}_{i,i+l}\right)\right) \tag{6}$$

where $log(\cdot)$ is the $SO(3)$ logarithm map, and the loss approximately equals $(log(\delta R_{i,i+l}\delta\widetilde{R}_{i,i+l}))^2/2$ for the small loss and $|log(\delta R_{i,i+l}\delta\widetilde{R}_{i,i+l})| - log(2)$ for the large loss.$\delta R_{i,i+l}$ and $\delta\widetilde{R}_{i,i+l}$ are the real orientation increments and the estimated orientation increments, respectively. Additionally, we also use a regularization loss as follows:

$$\mathcal{L}_\lambda = Max\left(\left|\widetilde{\omega}_i - \hat{\omega}_i\right|, \lambda\right) \tag{7}$$

where $\hat{\omega}_i$ is the raw gyroscope measurement, $\widetilde{\omega}_i$ is the de-noised gyroscope data, λ is a controllable parameter, only when the de-noised data deviates from the original measured value by more than a threshold value, it will punish the de-noised data to ensure the fast convergence of the network.

3.2.5. Implementation Details

For the training of our NGC-Net, we use a desktop environment equipped with Intel (R) Core(TM) i7-6700HQ 2.60 GHz CPU and NVIDIA GRID RTX8000-12Q 11 GB RAM. The framework of NGC-Net is implemented using PyTorch 1.5. The Adam optimizer [56] is used during training. We set the initial learning rate at 0.001 and adjust the learning rate adaptively. To prevent overfitting, the weight decay is set to 0.1 and the dropout parameter to 0.2. The model with the best validation loss is chosen as the best model for testing.

3.3. Muti-Sensor Fusion State Estimation

3.3.1. Image Processing

Similar to [11], we also use FAST [57] to detect feature points from each new image and track them between consecutive frames using the KLT sparse optical flow algorithm [58]. However, since the optical flow tracking method may result in lots of mismatches in complex environments, we conduct a reverse optical flow tracking method to reject the outliers, similar to VINS-fusion [13].

3.3.2. De-Noised IMU and Odometer Pre-Integration

In the pre-integration step, both the de-noised IMU pre-integration and odometer pre-integration need to be calculated at the same time to provide more constraints for system initialization and back-end nonlinear optimization. The pre-integration of IMU and odometer in discrete time is shown in (8):

$$\begin{cases} \hat{\alpha}_{i+1}^{I_k} = \hat{\alpha}_i^{I_k} + \hat{\beta}_i^{I_k}\delta t_i + \frac{1}{2}R\left(\hat{\gamma}_i^{I_k}\right)(\hat{a}_i - b_{a_i})\delta t_i^2 \\ \hat{\beta}_{i+1}^{I_k} = \hat{\beta}_i^{I_k} + R\left(\hat{\gamma}_i^{I_k}\right)(\hat{a}_i - b_{a_i})\delta t_i \\ \hat{\gamma}_{i+1}^{I_k} = \hat{\gamma}_i^{I_k} \otimes \begin{bmatrix} 1 \\ \frac{1}{2}\left(\widetilde{\omega}_i - b_{\omega_i}\right)\delta t_i \end{bmatrix} \\ \hat{\eta}_{i+1}^{I_k} = \hat{\eta}_i^{I_k} + R\left(\hat{\gamma}_i^{I_k}\right)R_o^I\hat{v}_i\delta t_i \\ b_{a_{i+1}} = b_{a_i} \\ b_{\omega_{i+1}} = b_{\omega_i} \\ (b_{\omega_i})_z = 0 \end{cases} \tag{8}$$

where i is the discrete moment corresponding to an IMU measurement within $[T_k, T_{k+1}]$, and T_k and T_{k+1} represent the corresponding time of the kth and $(k+1)$th image keyframes, respectively. δt_i is the time interval between two IMU measurements of i and $i+1$. $\hat{\alpha}^I$, $\hat{\beta}^I$, and $\hat{\gamma}^I$ denote the IMU position, velocity, and attitude pre-integration in the IMU coordinate system, respectively. $\hat{\eta}^I$ represents the position pre-integration of the odometer in the IMU coordinate system. At time $i = T_k$, $\hat{\alpha}_i^{I_k}$, $\hat{\beta}_i^{I_k}$, $\hat{\gamma}_i^{I_k}$, $\hat{\eta}_i^{I_k}$ are all zero. R_o^I is the extrinsic rotation matrix between the IMU sensor and the odometer. In this paper, the external parameters of the IMU and odometer are not corrected online; instead, we use the values calibrated offline directly. $\hat{a}_i$, $\widetilde{\omega}_i$, and $\hat{v}_i$ represent the accelerometer measurement, the de-noised gyroscope measurement, and the speed provided by wheel encoder reading at the time i. b_{a_i} and b_{ω_i} represent the bias of accelerometer and gyroscope at the time i.

Since our NGC-Net described in Section 3.2 can effectively reduce the noises of the gyroscope's original measurements, we use the de-noised gyroscope outputs instead of the raw gyroscope measurements to calculate these pre-integrations to achieve more accurate attitude, thus further improving the state estimation accuracy during the initialization and back-end nonlinear optimization.

It should be noted that to accurately calculate the IMU pre-integration and odometer pre-integration between the kth and $(k+1)$th image keyframes, we need to align the timestamps between image frames and IMU and odometer measurements. As implemented in [11], we interpolate the IMU measurements and wheel measurements based on the timestamps of the image frames and obtain all the IMU measurements and wheel measurements between the two image frames for calculating the IMU pre-integration and odometer pre-integration, respectively.

In the original VIO algorithm framework, to improve the accuracy of IMU input, IMU bias (including gyroscope bias) is calibrated online during the optimization stage. However, as demonstrated by [29–32] and our experiments in Section 4.3, we note that the horizontal direction integrated directly from the de-noised gyroscope can be more accurate than the results obtained by the state-of-the-art VIO methods. Thus, we always set $(b_{\omega_i})_z = 0$ when calculating the IMU and odometer pre-integration using the de-noised gyroscope measurements to maintain the best orientation accuracy.

The error state propagation equation of (8) at the discrete time is shown in Equation (9):

$$\begin{bmatrix} \delta\hat{\alpha}_{i+1}^{I_k} \\ \delta\hat{\beta}_{i+1}^{I_k} \\ \delta\hat{\theta}_{i+1}^{I_k} \\ \delta\hat{\eta}_{i+1}^{I_k} \\ \delta b_{a_{i+1}} \\ \delta b_{\omega_{i+1}} \end{bmatrix} = \begin{bmatrix} \mathbf{I} & \mathbf{I}\delta t_i & \mathbf{O} & \mathbf{O} & \mathbf{O} & \mathbf{O} \\ \mathbf{O} & \mathbf{I} & -R\left(\hat{\gamma}_i^{I_k}\right)\lfloor \hat{a}_i - b_{a_i} \rfloor_\times \delta t_i & \mathbf{O} & -R\left(\hat{\gamma}_i^{I_k}\right)\delta t_i & \mathbf{O} \\ \mathbf{O} & \mathbf{O} & \mathbf{I} - \left\lfloor \widetilde{\omega}_i - b_{\omega_i} \right\rfloor_\times \delta t_i & \mathbf{O} & \mathbf{O} & -\mathbf{I}\delta t_i \\ \mathbf{O} & \mathbf{O} & -R\left(\hat{\gamma}_i^{I_k}\right)\lfloor R_o^I \hat{v}_i \rfloor_\times \delta t_i & \mathbf{I} & \mathbf{O} & \mathbf{O} \\ \mathbf{O} & \mathbf{O} & \mathbf{O} & \mathbf{O} & \mathbf{I} & \mathbf{O} \\ \mathbf{O} & \mathbf{O} & \mathbf{O} & \mathbf{O} & \mathbf{O} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \delta\hat{\alpha}_i^{I_k} \\ \delta\hat{\beta}_i^{I_k} \\ \delta\hat{\theta}_i^{I_k} \\ \delta\hat{\eta}_i^{I_k} \\ \delta b_{a_i} \\ \delta b_{\omega_i} \end{bmatrix} + \begin{bmatrix} \mathbf{O} & \mathbf{O} & \mathbf{O} & \mathbf{O} & \mathbf{O} \\ -R\left(\hat{\gamma}_i^{I_k}\right)\delta t_i & \mathbf{O} & \mathbf{O} & \mathbf{O} & \mathbf{O} \\ \mathbf{O} & -\mathbf{I}\delta t_i & \mathbf{O} & \mathbf{O} & \mathbf{O} \\ \mathbf{O} & \mathbf{O} & -R\left(\hat{\gamma}_i^{I_k}\right)R_o^I\delta t_i & \mathbf{O} & \mathbf{O} \\ \mathbf{O} & \mathbf{O} & \mathbf{O} & \mathbf{I}\delta t_i & \mathbf{O} \\ \mathbf{O} & \mathbf{O} & \mathbf{O} & \mathbf{O} & \mathbf{I}\delta t_i \end{bmatrix} \begin{bmatrix} \mathrm{n}_a \\ \mathrm{n}_\omega \\ \mathrm{n}_v \\ \mathrm{n}_{b_a} \\ \mathrm{n}_{b_\omega} \end{bmatrix} = \mathbf{F}_i^{I_k}\delta\mathbf{z}_i^{I_k} + \mathbf{G}_i^{I_k}\mathbf{n} \tag{9}$$

where $\delta\hat{\theta}_i^{I_k}$ is the perturbation of rotation error, and its relationship with the rotation attitude is $\gamma_i^{I_k} \approx \hat{\gamma}_i^{I_k} \otimes \begin{bmatrix} 1 \\ \frac{1}{2}\delta\hat{\theta}_i^{I_k} \end{bmatrix}$. $\lfloor\cdot\rfloor_\times$ means the skew-symmetric matrix corresponding to a vector. $R\left(\hat{\gamma}_i^{I_k}\right)$ is the rotation matrix of attitude $\hat{\gamma}_i^{I_k}$. n_a,n_ω, and n_v represent the noises in accelerometer, gyroscope, and odometer measurements, respectively, which are commonly Gaussian white noises. n_{b_a} and n_{b_ω} denote the derivatives of accelerometer bias and gyroscope bias that are also white noises.

According to Equation (9), we can get the Jacobian matrix of $\hat{\alpha}^I$, $\hat{\beta}^I$, $\hat{\gamma}^I$, and $\hat{\eta}^I$ with respect to the IMU biases. The covariance matrix of each pre-integration term can also be obtained by forward propagation of the covariance. The calculation process is similar to [21], which is not explained in detail here.

3.3.3. System Initialization

During the initialization stage, the traditional monocular vision-inertial VIO system aims to recover the true scale of the visual map, the true scale velocity of each frame, the direction of gravity in the first IMU body frame, and the gyroscope biases. Thanks to our NGC-Net, we no longer need to solve the gyroscope bias in the initialization phase. In addition, the odometer can obtain real-scale speeds and positions combined with de-noised gyroscope measurements, helping to recover the real scale of monocular visual maps, i.e., obtaining the real-scale depth of map points. Therefore, in this initialization stage, our system only needs to solve the gravity direction and optimization velocity for each frame at the same time, which improves the speed of initialization.

For the consecutive kth frame and $(k+1)$th frame, we align the IMU pre-integration with the odometer pre-integration to construct the joint equation, as shown in Equations (10) and (11):

$$\hat{\alpha}_{I_{k+1}}^{I_k} = \hat{\eta}_{I_{k+1}}^{I_k} - v_{I_k}^{I_k}\delta T_k + P_o^I - R\left(\hat{\gamma}_{I_{k+1}}^{I_k}\right)P_o^I + \frac{1}{2}R\left(\hat{\gamma}_{I_0}^{I_k}\right)\mathrm{g}^{I_0}\delta T_k^2 \tag{10}$$

$$\hat{\beta}_{I_{k+1}}^{I_k} = R\left(\hat{\gamma}_{I_{k+1}}^{I_k}\right)v_{I_{k+1}}^{I_{k+1}} - v_{I_k}^{I_k} + R\left(\hat{\gamma}_{I_0}^{I_k}\right)\mathrm{g}^{I_0}\delta T_k \tag{11}$$

where δT_k means the time interval between image frames k and $k+1$, namely, $\delta T_k = T_{k+1} - T_k$. $v_{I_k}^{I_k}$ and $v_{I_{k+1}}^{I_{k+1}}$ represent the velocity of k and $k+1$ frames in the IMU body to be solved. g^{I_0} denotes the value of gravity in the first IMU body frame to be computed. P_o^I denotes the value of the extrinsic translation between the IMU and odometer, which is known because we directly use the values calibrated offline.

The equations constructed using every pair of consecutive frames in the sliding window are concatenated to establish linear simultaneous equations. By solving the linear

least-squares problem, the velocity of each frame and the gravity vector g^{I_0} in the first IMU frame can be solved.

Since the magnitude of gravity is usually a known quantity, the value used in this paper is $g = 9.81007$. To facilitate the iterative optimization of the gravity vector, we reset $\mathrm{g}^{I_0} = g\frac{\hat{\mathrm{g}}_i^{I_0}}{\left\|\hat{\mathrm{g}}_i^{I_0}\right\|} + \varphi_1 \mathrm{b}_1 + \varphi_2 \mathrm{b}_2$ and put it into (10) and (11). By resolving the new least-squares problem, the optimized gravity vector and the velocity of each frame are computed. Here, b_1 and b_2 are a pair of orthogonal bases in the tangent space of the gravity vector $\hat{\mathrm{g}}_i^{I_0}$. $\hat{\mathrm{g}}_i^{I_0}$ is the i^{th} iteration solution of g^{I_0}. φ_1 and φ_2 represent the values adjusted for the gravity vector $\hat{\mathrm{g}}_i^{I_0}$ along the b_1 and b_2 directions, which are solved from the new least-squares problem.

We repeat the iterative steps until $\hat{\mathrm{g}}^{I_0}$ convergence. In this way, we need not assume the vehicle moves on an approximately flat surface at this stage because we use the first solution computed from Equations (10) and (11) as the initial value for the subsequent optimization iterations.

Then, the first IMU frame is rotated to the world coordinate system. Finally, the optimized velocities and the IMU measurements are integrated to calculate the initial poses of all keyframes in the sliding window, and the feature points tracked between these frames are triangulated.

3.3.4. Nonlinear Optimization

The purpose of back-end nonlinear optimization is to combine a variety of measurements, such as camera, IMU and odometer, for high precision and robust estimation of vehicle states. To balance the requirements of computational load and real-time performance, similar to [11,12], we also use the sliding window with partial marginalization for nonlinear optimization.

The state vector to be estimated in the sliding window is defined as (12):

$$\begin{cases} \chi = \left[x_0,\ x_1,\ \ldots x_n, x_C^I, \lambda_0, \lambda_1 \ldots \lambda_m\right] \\ x_k = \left[p_{I_k}^w, v_{I_k}^w, q_{I_k}^w, b_{a_k}, b_{\omega_k}\right],\ k \in [0, n] \\ x_C^I = \left[p_C^I, q_C^I\right] \\ \left(b_{\omega_k}\right)_z = 0\ ,\ k \in [0, n] \end{cases} \tag{12}$$

where x_k represents the vehicle state at the time that the kth image is captured, including the position $p_{I_k}^w$, velocity $v_{I_k}^w$, orientation $q_{I_k}^w$ in the world frame, acceleration bias b_{a_k}, and gyroscope bias b_{ω_k} in the IMU body frame. n is the total number of keyframes in the sliding window. λ_l is the inverse depth of the lth feature from its first observation. x_C^I is the extrinsic parameters between the camera and IMU, including the relative position p_C^I and orientation q_C^I. We do not correct the extrinsic parameters between IMU and odometer but directly use the parameters calibrated offline.

At this stage, since the bias-corrected acceleration may provide more precise constraints for solving the pitch angle and roll angle, we further use the calibrated x-axis and y-axis biases of the gyroscope to update the pre-integration of IMU and odometer accordingly. However, as described in Section 3.3.2, since the learned horizontal direction can be more accurate than the result obtained by the state-of-the-art VIO methods (demonstrated by [29–32] and our experiments in Section 4.3), we do not update these pre-integrations using the z-axis bias of the gyroscope calibrated online; that is, we set $\left(b_{\omega_k}\right)_z$ to 0.

The objective function is composed of three residual terms, which are the visual reprojection residual, the IMU-odometer pre-integration residual, and the marginalization residual. The objective function is defined as follows:

$$\min_{\chi}\left\{\left\|r_p - H_p\chi\right\|^2 + \sum_{k\in[0,n-1]}\left\|r_\Omega\left(\hat{z}_{I_{k+1}}^{I_k}, \chi\right)\right\|_{P_{I_{k+1}}^{I_k}}^2 + \sum_{(l,j)\in\Psi}\rho\left(\left\|r_\Psi\left(\hat{z}_l^{c_j}, \chi\right)\right\|_{P_l^{c_j}}^2\right)\right\} \tag{13}$$

In Equation (13), the first term represents the marginalization residual, and $\{r_p, H_p\}$ is the prior information from marginalization. $r_\Omega\left(\hat{z}_{I_{k+1}}^{I_k}, \chi\right)$ is the IMU-odometer pre-integration residual, and Ω represents the set of IMU-odometer pre-integration in the sliding window. $P_{I_{k+1}}^{I_k}$ is the covariance matrix of IMU-odometer pre-integration from frame k to frame $k+1$. $r_\Psi\left(\hat{z}_l^{c_j}, \chi\right)$ is the visual reprojection residual. Ψ is the set of features and the corresponding frames in the current sliding window. l means the lth feature in Ψ, and c_j is the j^{th} image frame. $P_l^{c_j}$ is the uniform covariance matrix used for visual reprojection, and ρ is the robust kernel function.

Here, we give the combined residual function of IMU-odometer pre-integration, as shown in Equation (14):

$$r_\Omega\left(\hat{z}_{I_{k+1}}^{I_k}, \chi\right) = \begin{bmatrix} \delta\alpha_{I_{k+1}}^{I_k} \\ \delta\beta_{I_{k+1}}^{I_k} \\ \delta\theta_{I_{k+1}}^{I_k} \\ \delta\eta_{I_{k+1}}^{I_k} \\ \delta b_a \\ \delta b_\omega \end{bmatrix} = \begin{bmatrix} R_w^{I_k}\left(P_{I_{k+1}}^w - P_{I_k}^w + \frac{1}{2}\mathrm{g}^w\delta T_k^2 - v_{I_k}^w\delta T\right) - \hat{\alpha}_{I_{k+1}}^{I_k} \\ R_w^{I_k}\left(v_{I_{k+1}}^w + \mathrm{g}^w\delta T - v_{I_k}^w\right) - \hat{\beta}_{I_{k+1}}^{I_k} \\ 2\left[q_{I_k}^{w\,-1} \otimes q_{I_{k+1}}^w \otimes \left(\hat{\gamma}_{I_{k+1}}^{I_k}\right)^{-1}\right]_{xyz} \\ R_w^{I_k}\left(P_{I_{k+1}}^w - P_{I_k}^w\right) - P_o^I + R_w^{I_k}R_{I_{k+1}}^w P_o^I - \hat{\eta}_{I_{k+1}}^{I_k} \\ b_{a_{k+1}} - b_{a_k} \\ b_{\omega_{k+1}} - b_{\omega_k} \end{bmatrix} \tag{14}$$

Similar to [21], $\hat{\alpha}_{I_{k+1}}^{I_k}$, $\hat{\beta}_{I_{k+1}}^{I_k}$, $\hat{\gamma}_{I_{k+1}}^{I_k}$, and $\hat{\eta}_{I_{k+1}}^{I_k}$ are updated with IMU biases b_{a_k} and b_{ω_k}. Since we set $\left(b_{\omega_k}\right)_z = 0$, the z-axis bias of the gyroscope produces no updates to $\hat{\alpha}_{I_{k+1}}^{I_k}$, $\hat{\beta}_{I_{k+1}}^{I_k}$, $\hat{\gamma}_{I_{k+1}}^{I_k}$, and $\hat{\eta}_{I_{k+1}}^{I_k}$.

In this paper, Ceres Solver [59] is also used for solving this nonlinear problem.

3.3.5. Yaw Attitude Correction

The VINS system has 4 unobservable degrees of freedom (DOF) corresponding to 3 DOF of global translation and 1 DOF of rotation around the gravity vector (namely the yaw angle) [15]. Although the pre-integrations in Section 3.3.2 can provide relatively accurate residual constraints for the nonlinear optimization in Section 3.3.4, the yaw angle accuracy of the pose after optimization may still not be ideal.

In this stage, we further correct the yaw angle of each keyframe in the sliding window after nonlinear optimization to obtain the best attitude accuracy. Since the accuracy of attitude plays an important role in estimating the long-term trajectory, the corrected yaw angles are beneficial to improve the accuracy of our VIWO system. As shown in Section 4.4, we finally obtain more accurate trajectory results based on the accurate yaw attitude obtained by our NGC-Net.

For the correction steps, we first obtain the yaw angle of each keyframe in the sliding window by directly integrating the de-noised gyroscope measurements. Then, these yaw angles are substituted into the optimized poses of all keyframes in the sliding window. Finally, on the one hand, the corrected pose of the current keyframe is output as the final result. On the other hand, the corrected poses (besides the marginalized keyframe) in the sliding window are further used for the following keyframe optimization.

It is worth noting that we only substitute the yaw angle for the pose of each optimized keyframe during this stage, not including the pitch and roll angle.

4. Experiments

In this part, a series of experiments are carried out to verify the de-noising performance of our proposed NGC-Net and the accuracy and robustness of the multi-sensor fusion localization method assisted by the neural gyroscope calibration proposed in this paper.

4.1. Baselines

We compare our method with existing methods, including the following:

(1) Raw IMU: Orientation computed using the original IMU readings.
(2) OriNet: A 3D orientation estimation method based on an LSTM network [47].
(3) DIG-Net: Attitude estimation based on a dilated convolution network [29].
(4) Our proposed NGC-Net: Our learning-based method described in Section 3.2.
(5) VINS-Mono: Representative of state-of-the-art visual-inertial odometry with an open-source code [11].
(6) Open-VINS: A state-of-the-art filter-based visual-inertial estimator for which we choose the stereo and IMU configuration [7].
(7) Proposed VIWO: An optimization-based monocular visual-inertial-wheel odometry developed based on VINS-Mono without the aid of NGC-Net, which is similar to the work proposed by [11] without the online IMU-odometer extrinsic parameter calibration module.
(8) Proposed VIWO+NGC: A method that is the same as method (7), but the gyroscope inputs are the NGC-Net outputs rather than the raw gyroscope measurements.
(9) Proposed RNGC-VIWO: The method described in Section 3.3, in which the de-noised gyroscope measurements are effectively fused into the overall framework.

We divide the baseline methods into two categories corresponding to the two purposes of our experiments:

- 3D orientation estimates based on the learning method and VIO method. We compare these deep learning methods to demonstrate the de-noising performance of our NGC-Net. For a fair comparison, we use the same training sequence and test sequence. Since the OriNet has not been published and only provides test results on the EuRoC MAV Dataset [60], we do not compare our method with the OriNet on the KAIST Urban Dataset [61]. The DIG-Net is not tested on the KAIST Urban Dataset, although it is open-sourced, and we use the default network parameters and take the best training results as the network output. We also compare the VIO methods (VINS-Mono, Open-VINS) to demonstrate that our NGC-Net can accurately estimate the orientation and compete with VIO methods.
- 6DOF pose estimates based on multi-sensor fusion. We further evaluate the localization performance of our proposed RNGC-VIWO on the KAIST dataset, which is equipped with the stereo camera, IMU, and wheel odometer for vehicle localization. However, an open-source algorithm with the same sensor configuration cannot be found at present. Thus, we compare our RNGC-VIWO with the proposed VIWO, proposed VIWO+NGC, and the state-of-the-art VIO methods (VINS-Mono, Open-VINS). In addition, reference [37] proposes an excellent VIWO algorithm and tests it on KAIST urban39; we directly include the test results of urban39 in Section 4.4 for comparison since it is not open-source.

4.2. Metrics Definitions

We evaluate the method in terms of 3D orientation/yaw and 3D translation estimates, which are defined as follows:

(1) Absolute Yaw Error (AYE): The AYE computes the root mean square error between the ground truth and estimated heading as the following equation:

$$\mathrm{AYE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left\|\theta_i - \tilde{\theta}_i\right\|^2} \tag{15}$$

where n is the sequence length, and θ_i and $\tilde{\theta}_i$ are the ground truth and estimated yaw angle at the instant i.

(2) Absolute Orientation Error (AOE): The AOE calculates the root mean square error between the ground truth and estimated orientation as

$$\mathrm{AOE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left\|\log\left(R_i^T\widetilde{R}_i\right)\right\|_2^2} \tag{16}$$

where $\log(\cdot)$ is the $SO(3)$ logarithm map, n is the sequence length, and R_i and $\widetilde{R}_i$ are the ground truth and estimated orientation at the instant i.

(3) Absolute Translation Error (ATE): The ATE calculates the root mean square error between the ground truth and estimated position as

$$\mathrm{ATE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left\|P_i - \widetilde{P}_i\right\|^2} \tag{17}$$

where P_i and $\widetilde{P}_i$ are the ground truth and estimated position at the instant i.

4.3. Neural Gyroscope Calibration Network Performance

(1) Evaluation of the EuRoC MAV Dataset: The dataset provides stereo image and IMU data from a micro aerial vehicle including 11 sequences [60]. The IMU measurements are measured using the ADIS16448 at 200 Hz, and 6DOF pose ground truth is provided. As noticed in [7], the ground truth of the V1_01 easy sequence has incorrect orientation values, and we refer to the corrected ground truth provided by reference [7].

The same training and test sequences are used with the corresponding baseline methods [29,47] to make a fair comparison. The training set is defined as the first one and a half minutes of six sequences, namely V1_02_medium, V2_01_easy, V2_03_difficult MH_01_easy, MH_03_medium, and MH_05_difficult, and the validation set is the rest of these sequences. The test set contains five sequences of MH_02_easy, MH_04_difficult, V1_01_easy, V1_03_difficult, and V2_02_medium.

As shown in Table 1, our proposed NGC-Net is superior to the original IMU data in direction and yaw angle estimation. Compared with the other two learning methods, our proposed NGC-Net achieves the best performance on most test sequences, the AOEs and AYEs of which are 1.53/0.85 degrees. When comparing with VIO methods, we find that our NGC-Net is even comparable to the VIO method in orientation estimation. In addition, the mean yaw error of our NGC-Net is 0.85 degrees, which is better than all VIO methods such as Open-VINS with 1.37 degrees and VINS-Mono with 2.14 degrees. The orientation estimates and errors of the different methods on the test sequence of V1_01_easy and V1_03_difficult are also shown in Figure 4 and Figure 5, respectively.

Table 1. Absolute orientation error (AOE) and absolute yaw error (AYE) in terms of 3D orientation/yaw, in degrees. Results of VINS are the reported ones in [32]. The best results are in bold.

Sequence	Raw IMU	VINS-Mono [11]	Open-VINS [7]	OriNet [47]	DIG-Net [29]	NGC-Net
MH_02_easy	146/130	1.34/1.32	1.11/1.05	5.75/**0.51**	1.39/0.85	1.70/1.20
MH_04_difficult	130/77.9	1.44/1.40	1.60/1.16	8.85/7.27	1.40/0.25	**0.93/0.23**
V1_01_easy	71.3/71.2	0.97/0.90	0.80/0.67	6.36/2.09	1.13/0.49	**0.78/0.48**
V1_03_difficult	119/84.9	4.72/4.68	2.32/2.27	14.7/11.5	2.70/0.96	**1.05/0.75**
V2_02_medium	117/86.0	2.58/2.41	**1.85**/1.61	11.7/6.03	3.85/2.25	3.19/**1.57**
mean	125/89.0	2.21/2.14	**1.55**/1.37	9.46/5.48	2.10/0.96	**1.53/0.85**

In addition, the IMU of the EuRoc MAV Dataset runs at 200 Hz, the number of network model parameters is about 856,013, and the size of the network model is only 3.5 MB. The 1200 epochs of training take about 7 min.

(2) Evaluation of the KAIST Urban Dataset: This dataset contains a variety of sensor measurements, such as stereo camera images, IMU measurements, wheel encoder readings,

etc., which can provide the required inputs for our multi-sensor fusion localization system and is the main target dataset for our study. It also provides the ground truth generated by high-precision gyroscope, VRS-GPlS, and LiDAR sensors. The commercial-grade IMU measurements are obtained with Xsens MTi-300, and the frequency is 100 Hz. More details of the dataset are in [61].

In this experiment, we define the training set as 11 sequences, namely urban18, urban20, urban22, urban23, urban24, urban28, urban30, urban32, urban35, urban36, and urban38, and the validation set as the sequences of urban19, urban21, urban25, urban26, and urban27. The test set is urban29, urban31, urban33, urban34, urban37, and urban39.

The results of DIG-Net in Table 2 are the best results of dozens of training experiments using the default network parameters provided by [29], except for some parameters that must be adjusted to train on the KAIST Urban Dataset, such as IMU frequency, training sequences, etc.

Table 2. Absolute orientation error (AOE) and absolute yaw error (AYE) in terms of 3D orientation/yaw, in degrees. The best results are in bold, and the symbol -- fails to run on these test sequences.

Sequence	Raw IMU	VINS-Mono [11]	Open-VINS [7]	DIG-Net [29]	NGC-Net
urban29	37.64/26.47	3.15/2.82	**1.36**/0.81	3.33/**0.54**	2.91/0.69
urban31	93.68/60.74	16.02/15.72	--	8.39/6.98	**5.91/2.07**
urban33	84.98/75.94	10.25/9.93	3.61/3.37	6.12/3.69	**3.43/0.65**
urban34	43.95/40.75	24.38/24.08	--	2.57/2.16	**1.98/0.48**
urban37	48.23/29.99	67.27/5.57	7.34/6.29	3.24/2.22	**3.12/0.98**
urban39	122.34/120.48	19.97/19.94	**3.18**/2.95	7.46/5.60	5.11/**0.89**
mean	71.80/59.06	23.51/13.01	--	4.85/3.31	**3.74/0.96**

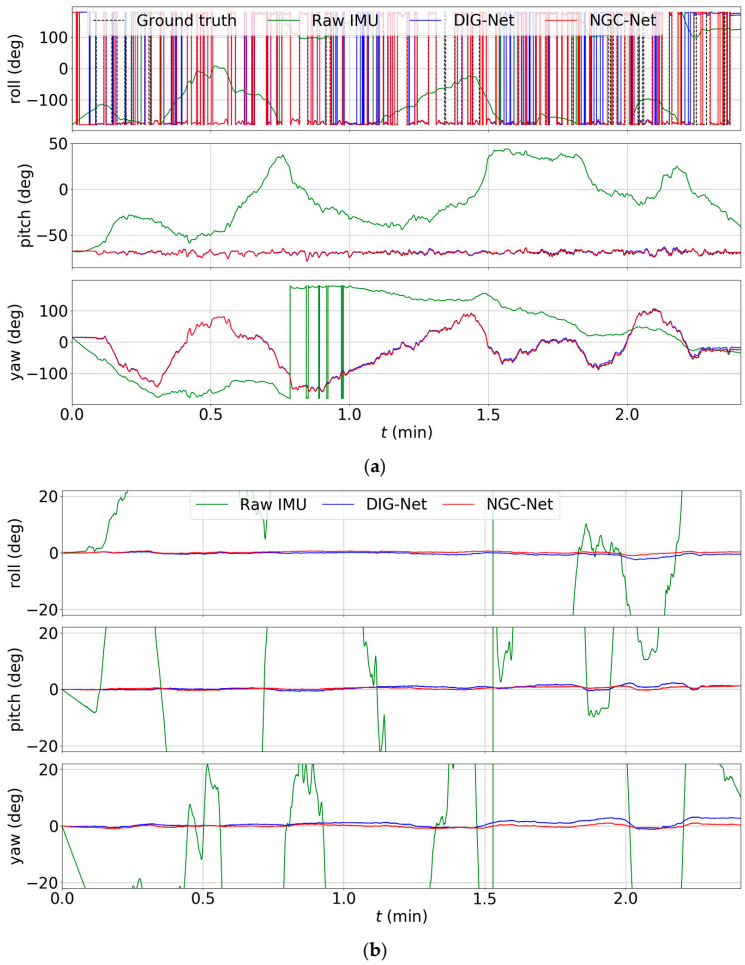
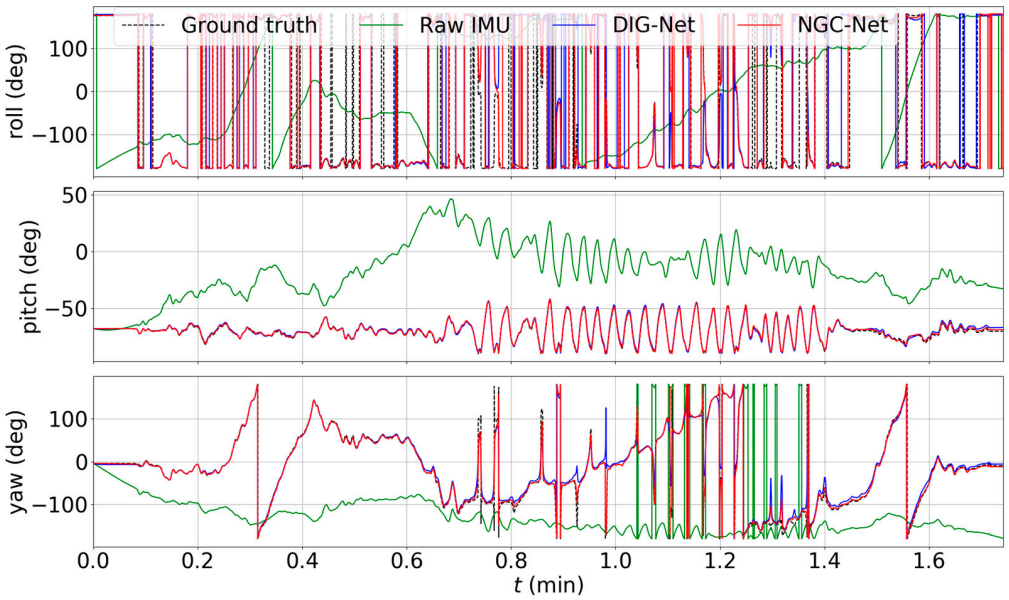
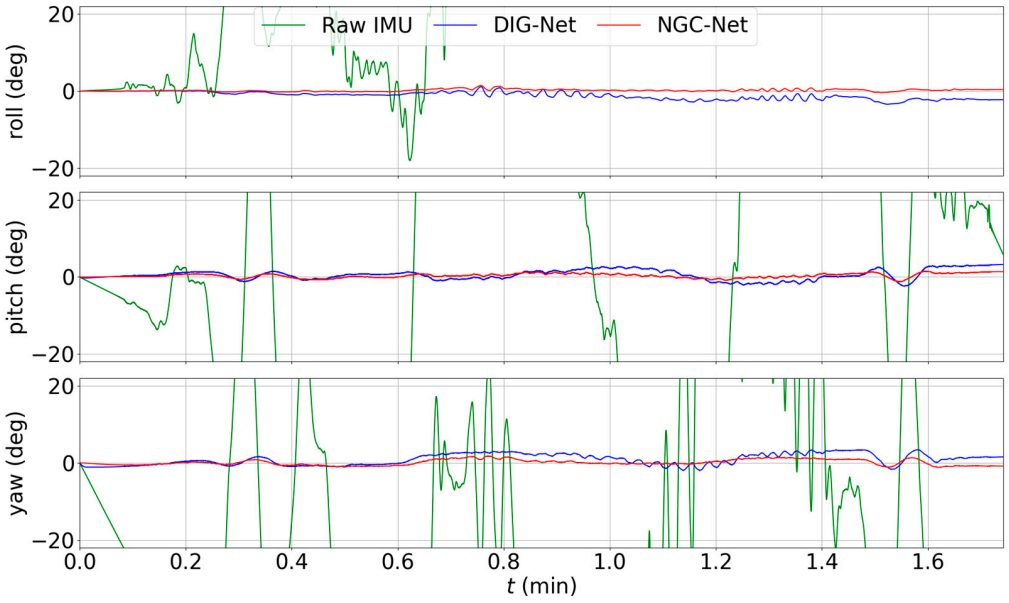


Figure 4. V1_01_easy with different methods: (**a**) estimated orientation; (**b**) 3D orientation errors.



(**a**)

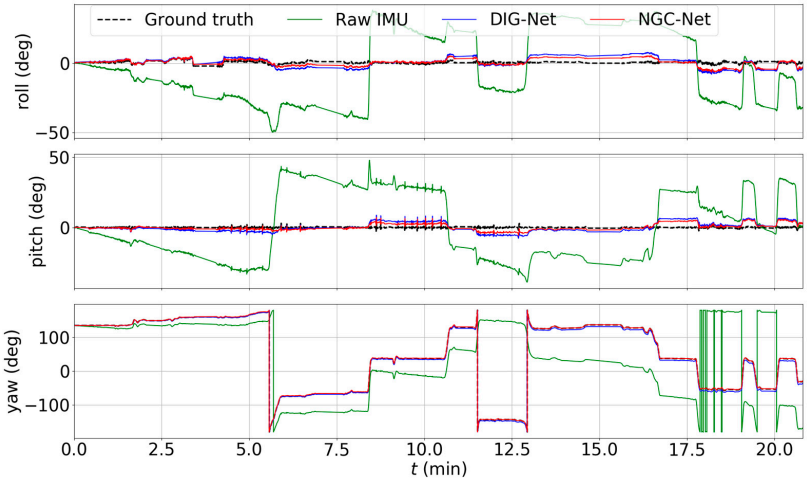


(**b**)

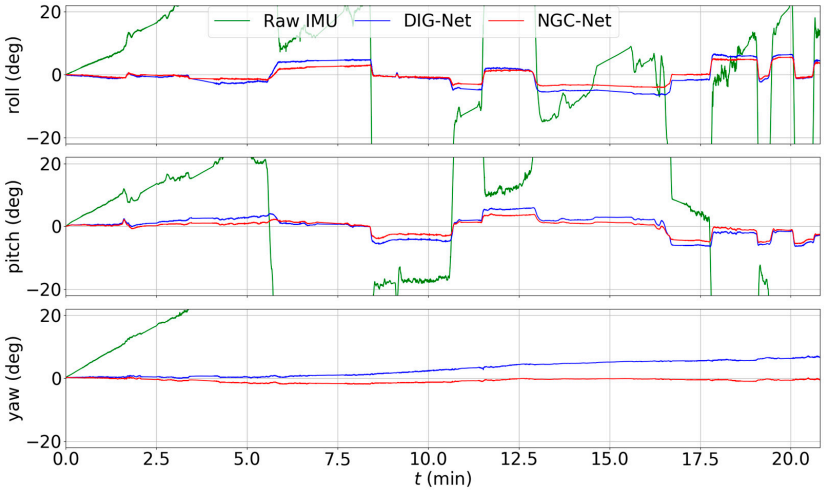
Figure 5. V1_03_difficult with different methods: (**a**) estimated orientation; (**b**) 3D orientation errors.

As shown in Table 2, in terms of 3D orientation and yaw angle estimation, our proposed NGC-Net outperforms the raw IMU data. Compared with DIG-Net, our NGC-Net maintains higher accuracy on most sequences except urban29. When comparing the estimated orientation of NGC-Net with VIO methods, the conclusion is similar to that of the EuRoC MAV Dataset. Compared with VINS-Mono, our proposed NGC-Net achieves the best performance on most test sequences, the AOEs and AYEs of which are 3.74/0.96 degrees. Compared with Open-VINS, which shows the excellent performance of its system, it still fails to run through the other two sequences since it still suffers from some initialization problems in complex dynamic environments. NGC-Net can provide more accurate 3D orientation estimation, especially accurate yaw angle estimation.

The orientation estimation and error of urban33 and urban39 by different methods are also shown in Figures 6 and 7, respectively. As shown in the figures, Our NGC-Net, the red line, is closest to ground truth and achieves the best performance among these methods.

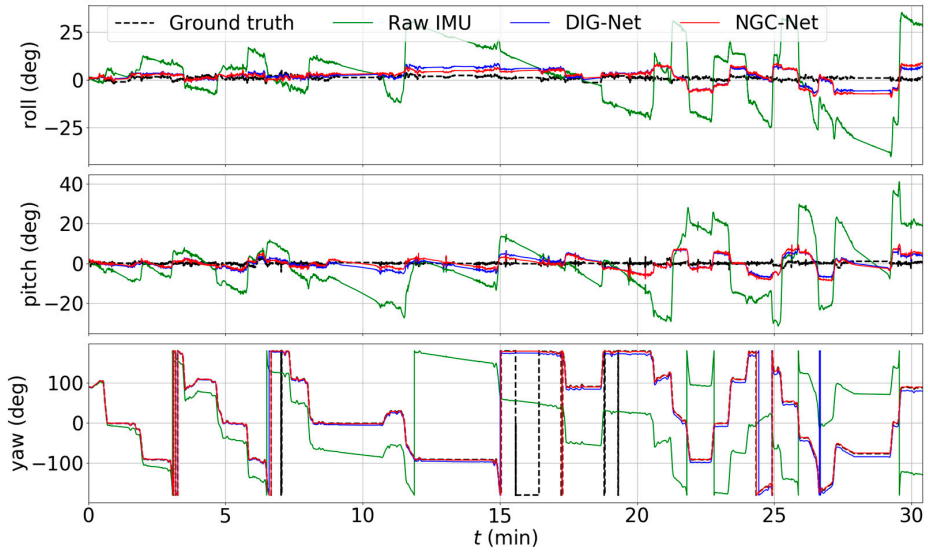


(**a**)



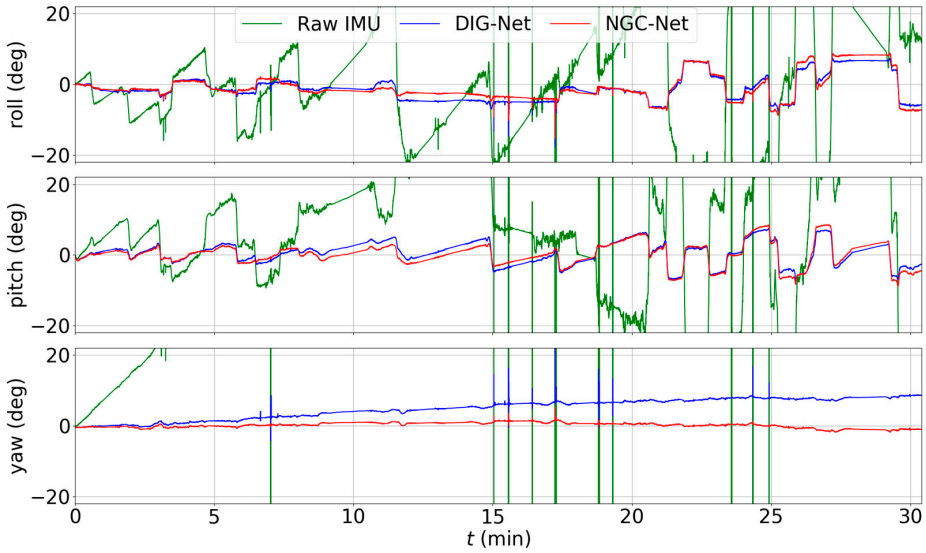
(**b**)

Figure 6. Urban33 with different methods: (**a**) estimated orientation; (**b**) 3D orientation errors.



(**a**)

Figure 7. *Cont.*



(**b**)

Figure 7. Urban39 with different methods: (**a**) estimated orientation; (**b**) 3D orientation errors.

In addition, the training time for 1200 epochs is approximately 24 min, and the size of the network model is about 3.5 MB. Moreover, the prediction time of the network model is about 10 s for the test sequence of more than 30 min and has real-time computing ability.

4.4. Muti-Sensor Fusion System Performance

To verify the performance of our approach in more realistic traffic scenarios, we use the KAIST Urban Dataset [61] to further evaluate the accuracy of our proposed multi-sensor fusion algorithm. The length of the dataset (3–12 km), the significant changes in illumination, and the complexity of the environment (including many pedestrians, oncoming traffic, high-rise buildings, etc.) lift it to another level of difficulty. We use the left of the stereo camera, the mounted commercial-grade IMU, and the left wheel encoder for pose estimation. Our proposed approaches are compared with the state-of-the-art methods such as Open-VINS [7], VINS-Mono [11], and VIWO [37] mentioned in Section 4.1. Since the purpose of our experiments is to evaluate the performance of these methods in complex long-term driving environments, we do not enable the loop-closing module for all these methods. Note that approaches [7,11] do not provide the test results on the KAIST Urban Dataset; we evaluate them ourselves by running each method five times on each testing sequence and taking the average values as the results. For reference [37], we only cite the results because it is not open-source.

Table 3 shows the root mean squared error (RMSE) of the 3D position (in units of meters) and 3D orientation (in units of degrees) of each algorithm. Figures 8 and 9 show the qualitative comparison of experimental results among different approaches. Figure 8 shows the comparison of estimated 2D trajectories of each tasting sequence among different approaches, and Figure 9 shows the detailed 6DOF trajectories of each test sequence among different approaches.

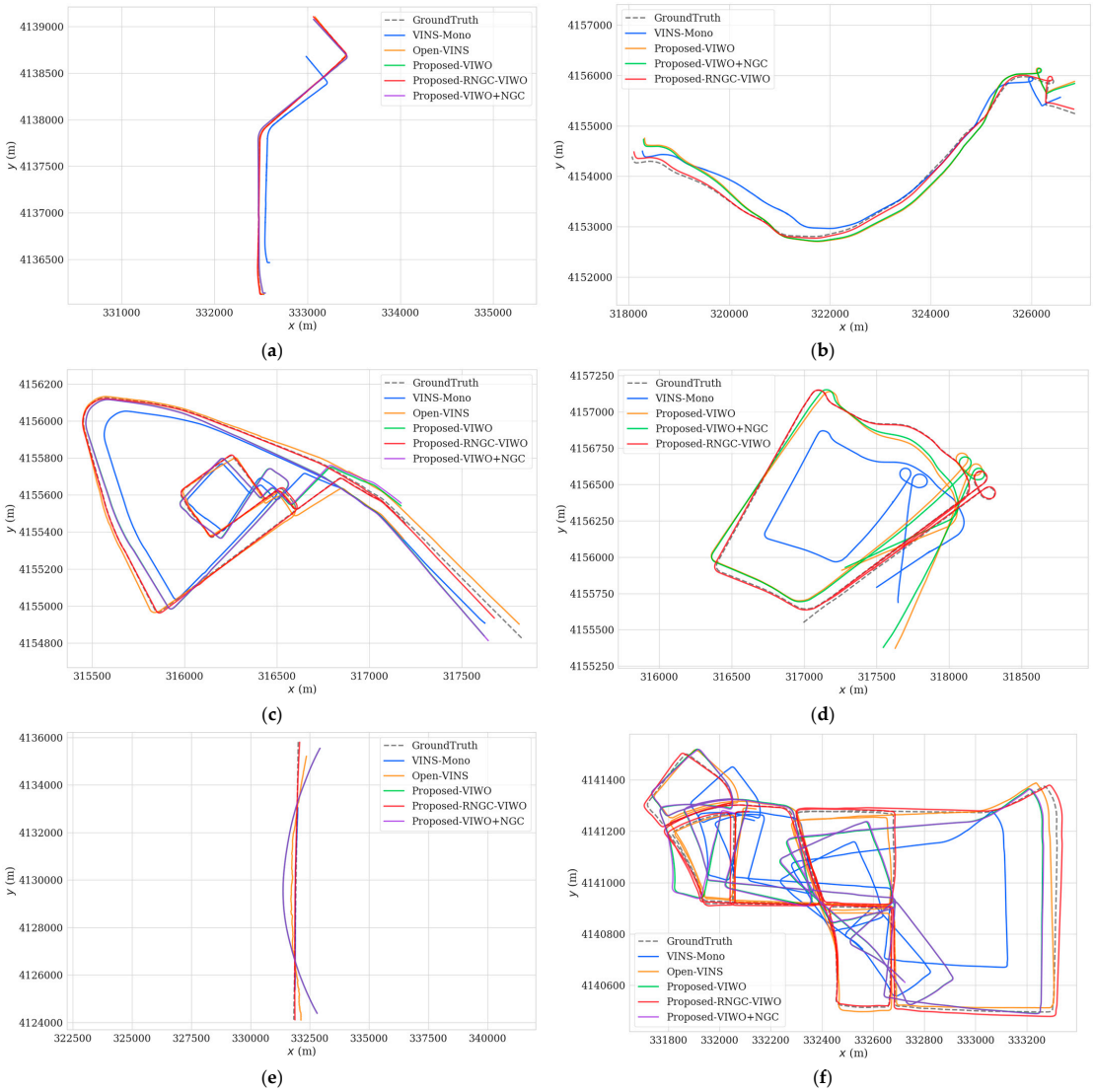


Figure 8. Visualization of 2D trajectory results on the six test sequences by different approaches: (**a**) urban29; (**b**) urban31; (**c**) urban33; (**d**) urban34; (**e**) urban37; (**f**) urban39.

Table 3. Comparison of absolute translation error (ATE) and absolute orientation error (AOE) in terms of 3D translation/orientation in units of meters/degrees, of resulting trajectories from different approaches. The best results are in bold.

Sequence	Length	VINS-Mono [11]	Open-VINS [7]	VIWO [37]	Proposed VIWO	Proposed VIWO+NGC	Proposed RNGC-VIWO
urban29	3.6 km	256.73/3.15	5.68/1.36	-	17.33/1.92	17.08/1.87	**4.06/0.84**
urban31	11.4 km	291.81/16.02	--	-	255.15/16.97	238.06/15.99	**49.83/2.99**
urban33	7.6 km	106.16/10.25	25.34/3.61	-	72.31/10.75	70.35/10.81	**11.74/1.55**
urban34	7.8 km	340.29/24.38	--	-	171.58/17.60	149.00/15.21	**17.60/2.37**
urban37	11.77 km	1282.76/67.27	182.51/7.34	-	463.97/40.50	463.84/40.25	**27.47/1.43**
urban39	11.06 km	153.70/19.97	24.12/3.18	52.65/2.87 * 42.74/**1.71** **	99.02/18.60	98.82/18.77	**20.11**/1.79
mean	8.87 km	380.80/23.51	--	-	179.89/17.72	172.86/17.15	**21.80/1.83**

It should be noted that the symbol * in Table 3 represents the result provided by [37] without online IMU-odometer spatiotemporal extrinsic calibration, and the symbol ** represents the result provided by [37] with online IMU-odometer spatiotemporal extrinsic calibration. The symbol - means the result is not provided by the method in [37], and the method is not open-source. The symbol -- in Table 3 means Open-VINS failed to run on these test sequences. In addition, we use the evo tools [62] to evaluate the 3D translation and orientation errors of these methods, using SE(3) Umeyama alignment in the calculation.

From Table 3 and Figures 8 and 9, we can see that our proposed RNGC-VIWO approach outperforms the state-of-the-art approaches in terms of both translation and orientation. From the above results, we can draw the following conclusions:

First, the pure VINS-Mono suffers from very large errors, up to hundreds of meters to about one kilometer, due to the scale drift affected by restricted motion (mostly constant speed, on straight lines) and complex environments. However, as more information becomes available, such as the addition of the right camera or the wheel encoder readings, which are methods of Open-VINS and our proposed VIWO, the positioning accuracy is significantly improved. Surprisingly, the 3D orientation accuracy of our proposed VIWO method is slightly improved compared with VINS-Mono, as the odometer pre-integration may provide some additional constraints for orientation estimation.

Second, by comparing our proposed VIWO and Open-VINS, the accuracy of Open-VINS in the four successfully initialized sequences is better than our proposed VIWO, but it still fails to run through the other two sequences in complex environments. Moreover, it should be noted that we choose the stereo-IMU configuration instead of the mono-IMU configuration of Open-VINS to test on the KAIST Urban Dataset.

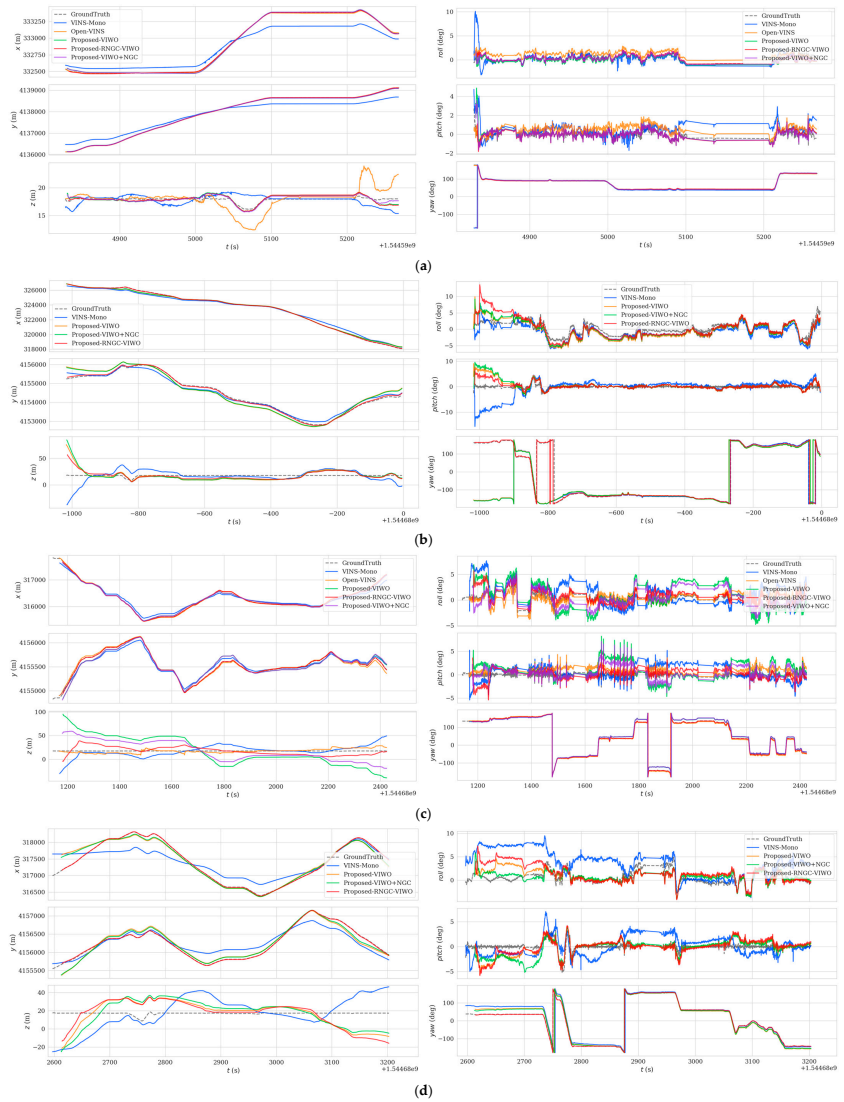


Figure 9. *Cont.*

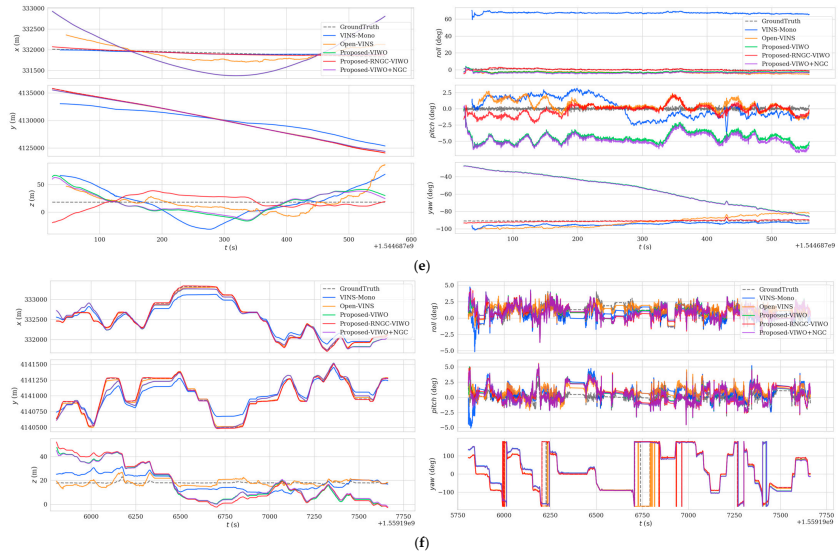


Figure 9. Trajectories of 6DOF results on the six test sequences by different approaches: (**a**) urban29; (**b**) urban31; (**c**) urban33; (**d**) urban34; (**e**) urban37; (**f**) urban39.

Third, in comparison with our proposed VIWO, our proposed VIWO+NGC approach just slightly improves the estimated position and orientation accuracy on the six test sequences, since it directly uses the de-noised gyroscope outputs as input. Comparing these two methods with our proposed RNGC-VIWO approach, our RNGC-VIWO can significantly improve the estimated position and orientation accuracy on the six test sequences, which proves the effectiveness of our proposed overall framework, which effectively fuses the de-noised gyroscope outputs into the traditional VIWO.

Fourth, our RNGC-VIWO approach also outperforms Open-VINS in terms of accuracy and robustness. Compared with the results on Urban39 provided by the VIWO method in reference [37], the position accuracy of our proposed RNGC-VIWO method still outperforms [37], even though [37] uses the stereo camera, IMU, and two wheel odometers along with online IMU-odometer spatiotemporal extrinsic calibration. The orientation accuracy of our proposed RNGC-VIWO on urban39 is comparable to the best result obtained by [37] with online IMU-odometer spatiotemporal extrinsic calibration and outperforms the result obtained by [37] without online IMU-odometer spatiotemporal extrinsic calibration. These results further prove the effectiveness of our proposed overall framework.

In conclusion, our proposed RNGC-VIWO approach can achieve only 21.80 m and 1.83-degree average errors over the long-term complex test sequences and a total length of 53.23 km for six sequences in the dataset, and the average distance of the six test sequences is 8.87 km, only using one camera, one wheel odometer, and a vehicle-mounted IMU de-noised by our NGC-Net.

We test these methods in the same Ubuntu 16.04 environment, using a desktop equipped with Intel (R) Core(TM) i7-6700HQ 2.60 GHz CPU and 16 GB RAM. Our proposed RNGC-VIWO method can achieve real-time performance similar to VINS-Mono [11], since the average time needed for visual feature detection and tracking is about 20 ms, and the maximum time of nonlinear optimization is an adjustable parameter (80 ms used in our experiments).

5. Conclusions

In this paper, to reduce the long-term drift of VIWOs deployed for ground vehicles without relying on additional GNSS sensors, prior maps, or specific motion patterns, we propose a tightly coupled nonlinear optimization method combining robust neural gyroscope observations with traditional visual-inertial-wheel odometry for pose estimation. The learning-based gyroscope calibration method can provide more powerful observations, and the de-noised gyroscope outputs can provide more accurate attitude estimation and

better directional constraints for the multi-sensor fusion method by effectively fusing them together.

We evaluate the proposed NGC-Net by comparing the orientation estimation performance with other learning-based methods on two public datasets, namely, the EuRoC MAV Dataset and the KAIST Urban Dataset. The results show that our NGC-Net achieves better de-noising performance compared with the existing learning-based methods and VIO methods and obtains excellent accurate attitude estimates with only a low-cost IMU. To verify the performance of our RNGC-VIWO in more realistic driving scenarios, we further use the KAIST Urban Dataset to evaluate the accuracy of our proposed method. The positioning accuracy of our RNGC-VIWO algorithm outperforms the state-of-the-art monocular and stereo visual-inertial methods, such as VINS-Mono and Open-VINS, and is even better than the stereo VIWO methods in [37]. The experimental results show our method has reliable performance in complex long-term environments.

In the future, we will further investigate how deep learning algorithms can be used to learn more sensor characteristics, such as accelerometers and odometers, and how to better integrate them with multi-sensor fusion algorithms to further improve localization accuracy. Moreover, deep learning for object detection has been the focus of much research and has been successful in many applications, including autonomous driving [63]. We also plan to incorporate object detection into our localization system to address dynamic interference and improve the robustness of vehicle localization in dynamic environments. Finally, inspired by [37], since online IMU-odometer spatiotemporal extrinsic calibration can improve the overall accuracy of the system, especially correcting the time offset between the IMU and odometer (0.027 s of the KAIST Dataset reported in [37]), we will add this online calibration module into our proposed framework to further improve the performance of vehicle localization.

Author Contributions: Conceptualization, M.Z. and C.D.; Data curation, H.T. and J.W.; Funding acquisition, B.L.; Methodology, M.Z., C.D. and H.Z.; Software, M.Z., C.D., H.T. and J.W.; Supervision, H.Z. and B.L.; Validation, M.Z. and C.D.; Writing—original draft, M.Z.; Writing—review and editing, M.Z. and C.D. All authors have read and agreed to the published version of the manuscript.

Funding: This research is funded by the project "the National Key Research and Development Program of China" (No. 2021YFB2501100).

Data Availability Statement: Our experimental data are all open-source datasets.

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

References

1. Gao, L.; Xiong, L.; Xia, X.; Lu, Y.; Yu, Z.; Khajepour, A. Improved Vehicle Localization Using On-Board Sensors and Vehicle Lateral Velocity. *IEEE Sens. J.* **2022**, *22*, 6818–6831. [CrossRef]
2. Klein, G.; Murray, D. Parallel tracking and mapping for small AR workspaces. In Proceedings of the 2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality, Nara, Japan, 13–16 November 2007; pp. 225–234.
3. Forster, C.; Pizzoli, M.; Scaramuzza, D. SVO: Fast semi-direct monocular visual odometry. In Proceedings of the 2014 IEEE International Conference on Robotics and Automation (ICRA), Hong Kong, China, 31 May–7 June 2014; pp. 15–22.
4. Wang, Y.; Zhang, S.; Wang, J. Ceiling-View Semi-DirectMonocular Visual Odometry with Planar Constraint. *Remote Sens.* **2022**, *14*, 5447. [CrossRef]
5. Mur-Artal, R.; Montiel, J.M.M.; Tardos, J.D. ORB-SLAM a versatile and accurate monocular SLAM system. *IEEE Trans. Robotics.* **2015**, *31*, 1147–1163. [CrossRef]
6. Wang, K.; Huang, X.; Chen, J.; Cao, C.; Xiong, Z.; Chen, L. Forward and Backward Visual Fusion Approach to Motion Estimation with High Robustness and Low Cost. *Remote Sens.* **2019**, *11*, 2139. [CrossRef]
7. Geneva, P.; Eckenhoff, K.; Lee, W.; Yang, Y.; Huang, G. Openvins: A research platform for visual-inertial estimation. In Proceedings of the 2020 IEEE International Conference on Robotics and Automation (ICRA), Paris, France, 31 May–31 August 2020; pp. 4666–4672.
8. Leutenegger, S.; Lynen, S.; Bosse, M.; Siegwart, R.; Furgale, P. Keyframe-based visual-inertial odometry using nonlinear optimization. *Int. J. Robot. Res.* **2015**, *34*, 314–334. [CrossRef]

9. Xu, B.; Chen, Y.; Zhang, S.; Wang, J. Improved Point–Line Visual–Inertial Odometry System Using Helmert Variance Component Estimation. *Remote Sens.* **2020**, *12*, 2901. [CrossRef]
10. Leutenegger, S.; Furgale, P.; Rabaud, V.; Chli, M.; Konolige, K.; Siegwart, R. Keyframe-Based Visual-Inertial SLAM using Nonlinear Optimization. In Proceedings of the Robotics: Science and Systems, Berkeley, CA, USA, 12–16 July 2014; pp. 789–795.
11. Qin, T.; Li, P.; Shen, S. Vins-mono: A robust and versatile monocular visual-inertial state estimator. *IEEE Trans. Robot.* **2018**, *34*, 1004–1020. [CrossRef]
12. Qin, T.; Pan, J.; Cao, S.; Shen, S. A General Optimization-based Framework for Local Odometry Estimation with Multiple Sensors. *arXiv* **2019**, arXiv:1901.03638.
13. Jiang, C.; Zhao, D.; Zhang, Q.; Liu, W. A Multi-GNSS/IMU Data Fusion Algorithm Based on the Mixed Norms for Land Vehicle Applications. *Remote Sens.* **2023**, *15*, 2439. [CrossRef]
14. Campos, C.; Elvira, R.; Rodríguez, J.J.G.; Montiel, J.M.M.; Tardós, J.D. ORB-SLAM3: An Accurate Open-Source Library for Visual, Visual–Inertial, and Multimap SLAM. *IEEE Trans. Robot.* **2021**, *37*, 1874–1890. [CrossRef]
15. Wu, K.J.; Guo, C.X.; Georgiou, G.; Roumeliotis, S.I. Vins on wheels. In Proceedings of the 2017 IEEE International Conference on Robotics and Automation (ICRA), Singapore, 29 May–3 June 2017; pp. 5155–5162.
16. Yu, Z.; Zhu, L.; Lu, G. Tightly-coupled Fusion of VINS and Motion Constraint for Autonomous Vehicle. *IEEE Trans. Veh. Technol.* **2022**, *14*, 5799–5810. [CrossRef]
17. Prikhodko, I.P.; Bearss, B.; Merritt, C.; Bergeron, J.; Blackmer, C. Towards self-navigating cars using MEMS IMU: Challenges and opportunities. In Proceedings of the 2018 IEEE International Symposium on Inertial Sensors and Systems (INERTIAL), Lake Como, Italy, 26–29 March 2018; pp. 1–4.
18. Ru, X.; Gu, N.; Shang, H.; Zhang, H. MEMS Inertial Sensor Calibration Technology: Current Status and Future Trends. *Micromachines* **2022**, *13*, 879. [CrossRef] [PubMed]
19. Gang, P.; Zezao, L.; Bocheng, C.; Shanliang, C.; Dingxin, H. Robust Tightly-Coupled Pose Estimation Based on Monocular Vision, Inertia and Wheel Speed. *arXiv* **2020**, arXiv:2003.01496.
20. Quan, M.; Piao, S.; Tan, M.; Huang, S.S. Tightly-Coupled Monocular Visual-Odometric SLAM Using Wheels and a MEMS Gyroscope. *IEEE Access* **2019**, *7*, 97374–97389. [CrossRef]
21. Liu, J.; Gao, W.; Hu, Z. Visual-Inertial Odometry Tightly Coupled with Wheel Encoder Adopting Robust Initialization and Online Extrinsic Calibration. In Proceedings of the 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Macau, China, 3–8 November 2019; pp. 5391–5397.
22. Li, T.; Zhang, H.; Gao, Z.; Niu, X.; El-sheimy, N. Tight Fusion of a Monocular Camera, MEMS-IMU, and Single-Frequency Multi-GNSS RTK for Precise Navigation in GNSS-Challenged Environments. *Remote Sens.* **2019**, *11*, 610. [CrossRef]
23. Gu, N.; Xing, F.; You, Z. Visual/Inertial/GNSS Integrated Navigation System under GNSS Spoofing Attack. *Remote Sens.* **2022**, *14*, 5975. [CrossRef]
24. Zhang, X.; Su, Y.; Zhu, X. Loop closure detection for visual SLAM systems using convolutional neural network. In Proceedings of the 23rd International Conference on Automation and Computing (ICAC), Huddersfield, UK, 7–8 September 2017; pp. 1–6.
25. Naseer, T.; Ruhnke, M.; Stachniss, C.; Spinello, L.; Burgard, W. Robust visual SLAM across seasons. In Proceedings of the 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Hamburg, Germany, 28 September–2 October 2015; pp. 2529–2535.
26. Schneider, T.; Dymczyk, M.; Fehr, M.; Egger, K.; Lynen, S.; Gilitschenski, I.; Siegwart, R. Maplab: An open framework for research in visual-inertial mapping and localization. *IEEE Robot. Autom. Lett.* **2018**, *3*, 1418–1425. [CrossRef]
27. Liu, W.; Caruso, D.; Ilg, E.; Dong, J.; Mourikis, A.I.; Daniilidis, K.; Kumar, V.; Engel, J. TLIO: Tight learned inertial odometry. *IEEE Robot. Autom. Lett.* **2020**, *5*, 5653–5660. [CrossRef]
28. Chen, C.; Lu, X.; Markham, A.; Trigoni, N. IoNet: Learning to cure the curse of drift in inertial odometry. In Proceedings of the 32th AAAI Conference on Artificial Intelligence 2018, New Orleans, LA, USA, 2–7 February 2018; pp. 6468–6476.
29. Brossard, M.; Bonnabel, S.; Barrau, A. Denoising imu gyroscopes with deep learning for open-loop attitude estimation. *IEEE Robot. Autom. Lett.* **2020**, *5*, 4796–4803. [CrossRef]
30. Liu, Y.; Liang, W.; Cui, J. LGC-Net: A Lightweight Gyroscope Calibration Network for Efficient Attitude Estimation. *arXiv* **2022**, arXiv:2209.08816.
31. Gao, Y.; Shi, D.; Li, R.; Liu, Z.; Sun, W. Gyro-Net: IMU Gyroscopes Random Errors Compensation Method Based on Deep Learning. *IEEE Robot. Autom. Lett.* **2023**, *8*, 1471–1478. [CrossRef]
32. Li, R.; Yi, W.; Fu, C.; Yi, X. Calib-Net: Calibrating the low-cost IMU via deep convolutional neural network. *Front. Robot. AI* **2021**, *8*, 772583. [CrossRef] [PubMed]
33. Xia, X.; Meng, Z.; Han, X.; Li, H.; Tsukiji, T.; Xu, R.; Zhang, Z.; Ma, J. Automated Driving Systems Data Acquisition and Processing Platform. *arXiv* **2022**, arXiv:2211.13425.
34. Mourikis, A.I.; Roumeliotis, S.I. A Multi-State Constraint Kalman Filter for Vision-Aided Inertial Navigation. In Proceedings of the 2007 IEEE International Conference on Robotics and Automation, Rome, Italy, 10–14 April 2007; pp. 3565–3572.

35. Zhang, Z.; Jiao, Y.; Huang, S.; Wang, Y.; Xiong, R. Map-based Visual-Inertial Localization: Consistency and Complexity. *arXiv* **2020**, arXiv:2204.12173.
36. Jung, J.H.; Cha, J.; Chung, J.Y. Monocular Visual-Inertial-Wheel Odometry Using Low-Grade IMU in Urban Areas. *IEEE Trans. Intell. Transp. Syst.* **2020**, *23*, 925–938. [CrossRef]
37. Lee, W.; Eckenhoff, K.; Yang, Y.; Geneva, P.; Huang, G. Visual-Inertial-Wheel Odometry with Online Calibration. In Proceedings of the 2020 International Conference on Intelligent Robots and Systems (IROS), Las Vegas, NV, USA, 24–30 October 2020; pp. 4559–4566.
38. Ghanipoor, F.; Hashemi, M.; Salarieh, H. Toward Calibration of Low-Precision MEMS IMU Using a Nonlinear Model and TUKF. *IEEE Sens. J.* **2020**, *20*, 4131–4138.
39. Jung, J.H.; Heo, S.; Park, C.G. Observability analysis of IMU intrinsic parameters in stereo visual–inertial odometry. *IEEE Trans. Instrum. Meas.* **2020**, *69*, 7530–7541. [CrossRef]
40. Liu, W.; Xiong, L.; Xia, X.; Lu, Y.; Gao, L.; Song, S. Vision-aided intelligent vehicle sideslip angle estimation based on a dynamic model. *IET Intell. Transp. Syst.* **2020**, *14*, 1183–1189. [CrossRef]
41. Xia, X.; Hashemi, E.; Xiong, L.; Khajepour, A. Autonomous Vehicle Kinematics and Dynamics Synthesis for Sideslip Angle Estimation Based on Consensus Kalman Filter. *IEEE Trans. Control Syst. Technol.* **2022**, *31*, 179–192. [CrossRef]
42. Liu, W.; Xia, X.; Xiong, L.; Lu, Y.; Gao, L.; Yu, Z. Automated Vehicle Sideslip Angle Estimation Considering Signal Measurement Characteristic. *IEEE Sens. J.* **2021**, *21*, 21675–21687. [CrossRef]
43. Xia, X.; Xiong, L.; Huang, Y.; Lu, Y.; Gao, L.; Xu, N.; Yu, Z. Estimation on IMU yaw misalignment by fusing information of automotive onboard sensors. *Mech. Syst. Signal Process.* **2021**, *162*, 107993. [CrossRef]
44. Clark, R.; Wang, S.; Wen, H.; Markham, A.; Trigoni, N. VINet: Visual-inertial Odometry as a Sequence-to-Sequence Learning Problem. In Proceedings of the AAAI Conference on Artificial Intelligence, San Francisco, CA, USA, 4–9 February 2017; pp. 3995–4001.
45. Chen, D.; Wang, N.; Xu, R.; Xie, W.; Bao, H.; Zhang, G. RNIN-VIO: Robust Neural Inertial Navigation Aided Visual-Inertial Odometry in Challenging Scenes. In Proceedings of the 2021 IEEE International Symposium on Mixed and Augmented Reality (ISMAR), Bari, Italy, 4–8 October 2021; pp. 275–283.
46. Chen, H.; Aggarwal, P.; Taha, T.M.; Chodavarapu, V.P. Improving Inertial Sensor by Reducing Errors using Deep Learning Methodology. In Proceedings of the NAECON 2018—IEEE National Aerospace and Electronics Conference, Dayton, OH, USA, 23–26 July 2018; pp. 197–202.
47. Esfahani, M.A.; Wang, H.; Wu, K.; Yuan, S. OriNet: Robust 3-D orientation estimation with a single particular IMU. *IEEE Robot. Autom. Lett.* **2019**, *5*, 399–406. [CrossRef]
48. Rehder, J.; Nikolic, J.; Schneider, T.; Hinzmann, T.; Siegwart, R. Extending kalibr: Calibrating the extrinsics of multiple IMUs and of individual axes. In Proceedings of the IEEE International Conference on Robotics and Automation, ICRA, Stockholm, Sweden, 16–21 May 2016; pp. 4304–4311.
49. Rohac, J.; Sipos, M.; Simanek, J. Calibration of low-cost triaxial inertial sensors. *IEEE Instrum. Meas. Mag.* **2015**, *18*, 32–38. [CrossRef]
50. Zhang, M.; Zhang, M.; Chen, Y.; Li, M. IMU data processing for inertial aided navigation: A recurrent neural network based approach. In Proceedings of the 2021 IEEE International Conference on Robotics and Automation (ICRA) 2021, Xi'an, China, 30 May–5 June 2021; pp. 3992–3998.
51. Bai, S.; Kolter, J.Z.; Koltun, V. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv* **2018**, arXiv:1803.01271.
52. Yu, F.; Koltun, V. Multi-Scale Context Aggregation by Dilated Convolutions. In Proceedings of the International Conference on Learning Representations 2016, San Juan, Puerto Rico, 2–4 May 2016.
53. Hendrycks, D.; Gimpel, K. Gaussian error linear units (GELUs). *arXiv* **2016**, arXiv:1606.08415.
54. Salimans, T.; Kingma, D.P. Weight normalization: A simple reparameterization to accelerate training of deep neural networks. *Adv. Neural Inf. Process. Syst.* **2016**, *29*, 901–909.
55. Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; Salakhutdinov, R. Dropout: A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.* **2014**, *15*, 1929–1958.
56. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.
57. Jianbo, S.; Tomasi. Good features to track. In Proceedings of the 1994 Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 21–23 June 1994; pp. 593–600.
58. Lucas, B.D.; Kanade, T. An Iterative Image Registration Technique with an Application to Stereo Vision. In Proceedings of the 7th International Joint Conference on Artificial Intelligence, IJCAI'81, San Francisco, CA, USA, 24–28 August 1981; pp. 674–679.
59. Agarwal, S.; Mierle, K.; The Ceres Solver Team. Ceres Solver. Available online: http://ceres-solver.org (accessed on 1 March 2022).
60. Burri, M.; Nikolic, J.; Gohl, P.; Schneider, T.; Rehder, J.; Omari, S.; Achtelik, M.W.; Siegwart, R. The EuRoC micro aerial vehicle datasets. *Int. J. Robot. Res.* **2016**, *35*, 1157–1163. [CrossRef]
61. Jeong, J.; Cho, Y.; Shin, Y.S.; Roh, H.; Kim, A. Complex urban dataset with multi-level sensors from highly diverse urban environments. *Int. J. Robot. Res.* **2019**, *38*, 642–657. [CrossRef]

62. Grupp, M. Evo: Python Package for the Evaluation of Odometry and Slam. Available online: https://github.com/MichaelGrupp/evo (accessed on 10 March 2022).
63. Liu, W.; Quijano, K.; Crawford, M.M. YOLOv5-Tassel: Detecting tassels in RGB UAV imagery with improved YOLOv5 based on transfer learning. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2022**, *15*, 8085–8094. [CrossRef]