IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received May 16, 2020, accepted May 26, 2020, date of publication June 8, 2020, date of current version July 7, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3000555

SIMSF: A Scale Insensitive Multi-Sensor Fusion Framework for Unmanned Aerial Vehicles Based on Graph Optimization

BO DAI[1,2,3], YUQING HE[1,2], (Member, IEEE), LIYING YANG[1,2], (Member, IEEE), YUN SU[1,2,3], YUFENG YUE[4], (Member, IEEE), AND WEILIANG XU[5], (Senior Member, IEEE)

[1]State Key Laboratory of Robotics, Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang 110016, China
[2]Institutes for Robotics and Intelligent Manufacturing, Chinese Academy of Sciences, Shenyang 110016, China
[3]University of Chinese Academy of Sciences, Beijing 100049, China
[4]School of Electrical and Electronics Engineering, Nanyang Technological University, Singapore 639798
[5]Department of Mechanical Engineering, The University of Auckland, Auckland 1010, New Zealand

Corresponding author: Yuqing He (heyuqing@sia.cn)

This work was supported in part by the National Key Research and Development Program of China under Grant 2017YFC0822201, in part by the National Natural Science Foundation of China under Grant 91648204 and Grant U1608253, and in part by the Autonomous Control Technology of Heterogeneous Intelligent System under Grant 41412040202.

ABSTRACT Given the payload limitation of unmanned aerial vehicles (UAVs), lightweight sensors such as camera, inertial measurement unit (IMU), and GPS, are ideal onboard measurement devices. By fusing multiple sensors, accurate state estimations can be achieved. Robustness against sensor faults is also possible because of redundancy. However, scale estimation of visual systems (visual odometry or visual inertial odometry, VO/VIO) suffers from sensor noise and special-case movements such as uniform linear motion. Thus, in this paper, a scale insensitive multi-sensor fusion (SIMSF) framework based on graph optimization is proposed. This framework combines the local estimation of the VO/VIO and global sensors to infer the accurate global state estimation of UAVs in real time. A similarity transformation between the local frame of the VO/VIO and the global frame is estimated by optimizing the poses of the most recent UAV states. In particular, for VO, an initial scale is estimated by aligning the VO with the IMU and GPS measurements. Moreover, a fault detection method for VO/VIO is also proposed to enhance the robustness of the fusion framework. The proposed methods are tested on a UAV platform and evaluated in several challenging environments. A comparison between our results and the results from other state-of-the-art algorithms demonstrate the superior accuracy, robustness, and real-time performance of our system. Our work is also a general fusion framework, which can be extended to other platforms as well.

INDEX TERMS Multi-sensor fusion, graph optimization, fusion framework, scale insensitive, unmanned aerial vehicle, state estimation.

I. INTRODUCTION

Unmanned aerial vehicles (UAVs) have been widely applied in military and civilian fields. Due to their maneuverability, vertical take-off and landing (VTOL) capability, and easy manipulation, they have been utilized in emergency search and rescue efforts, security monitoring, and aerial photography [1], [2]. During flight, the most fundamental element is state estimation, which includes the position, velocity, and attitude of the UAV. These states can be monitored by complementary sensors, such as GPS that provides position information and inertial measurement unit (IMU) that provides attitude information. However, compared to this complementary method of state estimation, multi-sensor fusion can achieve more accurate state estimations. Due to the redundancy it generates, robustness against sensor faults can be achieved. For example, even if one sensor produces a fault, the state estimator can still work, but with lower accuracy.

There are many sensors on UAVs used for state estimation, such as GPS, light detection and ranging (LiDAR) device,

The associate editor coordinating the review of this manuscript and approving it for publication was Juan A. Lara.

camera and IMU [3]. These sensors can be divided into two types. The first type is local sensors, such as camera and LiDAR. By matching several of the latest frames from the measurements, the current state can be estimated iteratively. There are a number of effective methods for local pose estimation, such as LiDAR-based odometry [4], [5] and visual-based odometry [6]–[8]. Among these, [6], [7] could not recover metric scale information because only a monocular camera was utilized. Thus, in [9]–[11], visual inertial-based odometry methods were proposed. These algorithms can achieve precise 6-DoF (Degrees of Freedom) estimation in a small region. They are well suited to cases that do not require global information, such as hovering and auto-stable flight. However, there are obvious drawbacks to these odometry methods. First, the estimated state refers to a local coordinate (the initial point) rather than the global coordinate. When the initial point changes, a different estimation will be acquired even in the same place. Thus, some tasks that require global information are unachievable. Second, the current state is estimated by optimizing several of the latest frames, which is a process that accumulates estimation errors due to noise and a lack of global measurements. Additionally, in special conditions such as uniform linear motion, the scale of visual odometry (VO) and visual inertial odometry (VIO) may drift or become unobservable. These errors can be eliminated by adding a loop closure module [12], [13], but during large-scale flight, it will consume large amounts of computing power and sacrifice real-time performance. Moreover, loop closure may not exist at all during the entire flight. The second type is global sensors, such as GPS, barometer, and magnetometer. These sensors provide drift-free measurements, but normally they are noisy and easily being interfered. For example, GPS is very sensitive to weather conditions, electromagnetic radiation, and occlusion by buildings. Thus, how to fuse these two types of sensors to achieve locally accurate and globally drift-free estimation has been a priority among researchers. Specially for UAV, it raises higher requirements such as real-time performance and robustness. There are several algorithms for multi-sensor fusion, such as the filter-based fusion method [10], [14]–[16]. This method can fuse local visual information with data from global sensors, but it needs an accurate initial transformation between local and global coordinates and it assumes that the transformation will not drift. Another method is the optimization-based fusion method [17]–[22], which optimizes a sliding window pose graph consisting of current and previous states. Due to the fact that the states are overly constrained, the optimization-based method more effectively manages drifts, such as position and scale drifts, than the filter-based method.

In this paper, a scale insensitive multi-sensor fusion framework for UAV based on graph optimization is proposed, which can process multiple sensor measurements including GPS, IMU, VO/VIO, barometer and magnetometer. The main problem of multi-sensor fusion is how to transfer the states in the local frame of the VO/VIO into the global frame. Thus, a similarity transformation between these two frames is estimated by executing a sliding window graph optimization, which is the key for achieving scale insensitivity. Specifically for VO input, a fast and effective scale initialization method that aligns VO with online GPS and IMU measurements, even when the UAV is still flying, is proposed. In addition, a simple fault detection method is proposed to enhance the robustness against sensor faults. The main contributions of this paper are as follows:

- A scale insensitive multi-sensor fusion (SIMSF) framework based on graph optimization that can achieve locally accurate and globally drift-free state estimations.
- A scale insensitive initialization for VO input that can provide an initial scale of VO even during flight.
- A fault detection module that can determine whether VO/VIO is incorrect and needs to restart.
- Experimental validation and evaluation of the proposed methods that demonstrate the improvement in accuracy, robustness, and real-time measurements compared to other state-of-the-art multi-sensor fusion systems.

II. RELATED WORK

In recent years, state estimation using multi-sensor fusion methods has been generating interest due to its higher accuracy and robustness. These fusion methods can be categorized into two main types: filter-based and optimization-based.

Filter-based methods are normally based on maximum a posterior estimations, such as the Kalman Filter (KF) and its extended (EKF) and unscented (UKF) variants. Weiss *et al.* [14], [15] constructed a general modular multi-sensor fusion framework to process IMU, GPS, and vision measurements. In both cases, the propagation is driven by a high frequency IMU, and other sensors are aligned with it. Moreover, sensor delays are also considered in modular design to fuse sensors with different frequencies. Bloesch *et al.* [10] introduced an intensity error into the EKF and utilized a tight-couple fusion method to obtain a VO. Due to the intensity error, the system has a better robustness against disturbance. Shen *et al.* [16] proposed a fusion framework based on the UKF to fuse vision, LiDAR, and GPS information. That framework can achieve accurate state estimations in both indoor and outdoor environments. Compared with the EKF, the UKF does not require linearization of the dynamics nor the continuity of the observation, and the power consumption of the computations increases linearly with the dimension. Filter-based methods enable modular designs to manage the synchronization and delay between different sensors, which guarantees real-time performance. However, filter-based methods normally assume that the states satisfy the first-order Markov hypothesis, the premise of which is that the current state is related only to the previous state. Thus, older states are superfluous for estimating the current state. Filter-based methods are advantageous when processing global sensors, but perform suboptimally compared to optimization-based methods [23].

In contrast, optimization-based methods assume that current states satisfy the multi-order Markov hypothesis and

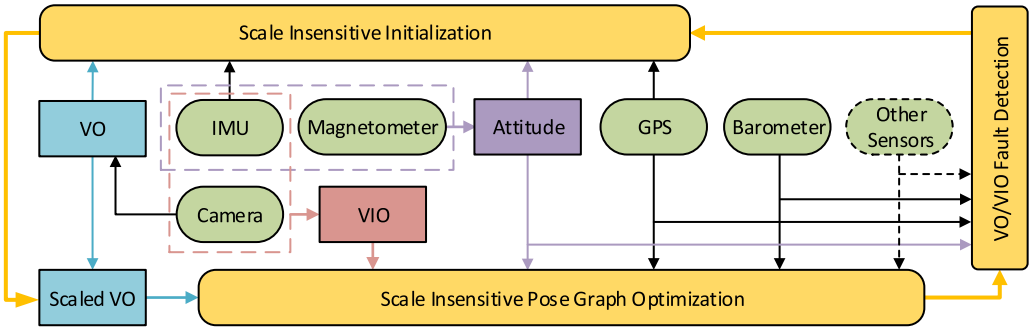

FIGURE 1. Illustration of the proposed SIMSF framework, which includes a scale insensitive initialization module, a scale insensitive pose graph optimization module, and a VO/VIO fault detection module.

that all past states can be used to estimate current states. There are a number of effective optimization-based algorithms that obtain robust and scale-known VIO, such as OKVIS [9] and VINS-Mono [11]. For global sensor fusion, Rehder *et al.* [17] achieved global pose estimation by fusing GPS and stereo vision. This method improved the estimation accuracy in a large scale environment by using the relation between multiple frames rather than two images from stereo. Merfels and Stachniss [18] proposed a sliding window method to fuse VIO and GPS measurements by optimizing the states in the window, and also proposed a corresponding marginalization method. Similar work can also be found in GOMSF [20], but the difference between GOMSF and [18] is that GOMSF estimated the transformation between the local frame of the VIO and the global frame to achieve a higher frequency estimation output rather than estimating the current state directly. In [19], [21], a prior map was needed for localization with a weak GPS environment. The stability of the system was ensured by decoupling the local VIO from the global registration to the reference map. Qin *et al.* [22] proposed a multi-sensor fusion framework, VINS-Fusion, that treats each sensor as a constraint factor and fuses the readings from GPS, barometer and other sensors. However, scale drift, which decreases the accuracy of state estimation, was not considered in the above studies.

III. SYSTEM OVERVIEW

The structure of the proposed SIMSF framework is shown in Fig. 1. There are three levels in the framework: the sensor level (bottom level, shown in green), the processing level (middle level, shown as squares), and the fusion level (top level, shown in yellow).

The bottom sensor level includes all relevant sensors, including IMU, magnetometer, camera, GPS, barometer, and other sensors potentially used. IMU, which contains accelerometer and gyroscope, provides acceleration and angular velocity measurements. Magnetometer measures the earth's magnetic field with respect to the frame of the UAV body. Combining that with the local magnetic declination, heading information can be extracted. Camera provides visual measurements such as intensity and texture. GPS produces calculations of longitude, latitude, and altitude, which can be converted to a position in 3D Euclidean space by setting an initial GPS point. Barometer is used to measure air pressure, which can be converted to height information, but normally height measurement is very noisy. There are many other potentially useful sensors that provide distance information, such as lightweight LiDAR or sonar, and velocity information, such as optical flow sensors.

The middle level processes the measurements from the sensor level. There are three modules in the middle level: attitude, VIO, and VO. The attitude module is used to fuse the IMU and magnetometer measurements to obtain accurate attitude information via a simple complementary filter, as proposed in [24]. The other two modules are both odometry, which can provide 6-DoF state estimations. VIO fuses IMU and camera measurements using the existing algorithm VINS-Mono [11], which can provide a metric scale in ideal conditions. Moreover, to enhance the expansibility of the framework, VO is also considered as a possible input of the system. By combining it with the scale estimated by the scale insensitive initialization module, a scaled VO is generated. In this paper, VO is chosen as the output of ORB-SLAM [13].

The top fusion level is divided into three main parts: scale insensitive initialization, scale insensitive pose graph optimization, and VO/VIO fault detection. The scale insensitive initialization module primarily estimates the scale of VO by aligning it with IMU and global pose measurements (provided by attitude and GPS), which is the foundation for using VO-type input. The scale insensitive pose graph optimization module is used to align VIO or scaled VO with attitude, GPS, and barometer measurements by optimizing a pose graph window containing the most recent states. The output of the optimization is the similarity transformation between the local frame of VIO or scaled VO with the global frame, which leads to a VO/VIO-rate global state output. The VO/VIO fault detection module runs in parallel by checking the motion between different types of sensors.

Here we define the notation and frame definitions used throughout this paper. The world frame or global frame, also known as the East-North-Up (ENU) frame, is denoted by $(.)^w$. The body frame, the first axis of which is defined as the forward direction of the UAV, is denoted by $(.)^b$. The local frame of VO/VIO is represented by $(.)^l$, and the camera frame

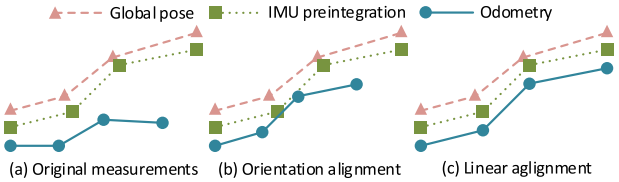

FIGURE 2. Illustration of scale insensitive initialization, including orientation alignment and linear alignment.

is represented by $(.)^c$. The rotation matrices $\mathbf{R}$ and Hamilton quaternions $\mathbf{q}$ both represent rotations, which also can be represented by transformations $\mathbf{T}$. Translation and rotation from the body frame to the world frame are represented by $\mathbf{p}_b^w$ and $\mathbf{q}_b^w$, respectively. Translation, rotation, and scaling from the VO/VIO frame to the body frame are denoted by $\mathbf{p}_c^b$, $\mathbf{q}_c^b$, and s, respectively, and are also described by a similarity transformation $\mathbf{S}_c^b$. The state while taking the kth image is represented by $(.)_k$. Multiplication between quaternions is denoted by $\otimes$. Original sensor measurements, including noise, are represented by $\hat{(.)}$. Finally, $\mathbf{g}^w = [0, 0, g]^{\mathrm{T}}$ is the gravity vector in the world frame.

IV. METHODOLOGY

A. SCALE INSENSITIVE INITIALIZATION

The main purpose of this module is to estimate a rough initial similarity transformation from the local frame l of the VO to the global frame w. There are also two special cases that need this module. The first one is when the input is of an unknown type (i.e., it is not known in advance whether the input is VO, VIO, or another type of odometry such as LiDAR). Another case is when initialization is needed during flight, such as when the VIO system is restarted after it faults. In this case, normally the UAV is flying in uniform linear motion, which prevents the scale of VIO from being observable [25]. The idea here is motivated by the initialization of VINS-Mono, which aligns a structure from motion (SfM) result with IMU pre-integration. Here we use odometry instead of SfM, and utilize global attitude and GPS measurements as well as IMU readings.

Given that IMU pre-integration and global pose are both utilized in the initialization, similarity matrices $\mathbf{S}_c^b$ and $\mathbf{S}_l^w$ are both needed. The frames l and c have the same scale, and the frames w and b also have the same scale. We have

$$\begin{aligned} \mathbf{S}_c^b &= \begin{bmatrix} s\mathbf{R}_c^b & \mathbf{p}_c^b \\ \mathbf{0} & 1 \end{bmatrix} \\ \mathbf{S}_l^w \mathbf{T}_{c_k}^l &= \mathbf{T}_{b_k}^w \mathbf{S}_c^b, \end{aligned} \tag{1}$$

where $\mathbf{T}_{b_k}^w$ can be obtained directly via the attitude and GPS measurements and $\mathbf{T}_{c_k}^l$ represents the current camera pose in the local frame l provided by the VO. To obtain $\mathbf{S}_l^w$, rotation matrices $\mathbf{R}_c^b$ and $\mathbf{R}_l^w$ and the scale s are needed. An illustration of the initialization process is shown in Fig. 2. First, the orientation of the odometry is aligned with the IMU pre-integration and global pose (derived from the attitude and GPS module) to solve for $\mathbf{R}_c^b$ and $\mathbf{R}_l^w$. Second, a linear alignment of these three types of measurements is performed to obtain the metric scale s of the odometry.

1) IMU PREINTEGRATION

We now introduce the basic theory of IMU pre-integration. Given that IMU is fixed in the body frame, acceleration $\mathbf{a}$ and angular velocity $\hat{\boldsymbol{\omega}}$ measurements can be modeled by

$$\begin{aligned} \hat{\mathbf{a}}_t &= \mathbf{a}_t + \mathbf{b}_{a_t} + \mathbf{R}_w^t \mathbf{g}^w + \mathbf{n}_a \\ \hat{\boldsymbol{\omega}}_t &= \boldsymbol{\omega}_t + \mathbf{b}_{\omega_t} + \mathbf{n}_\omega, \end{aligned} \tag{2}$$

where $\mathbf{n}_a$ and $\mathbf{n}_\omega$ represent the additive noise of acceleration and angular velocity, respectively. Both are Gaussian white noise, meaning $\mathbf{n}_a \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\sigma}_a^2)$ and $\mathbf{n}_\omega \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\sigma}_\omega^2)$. In addition, the bias of acceleration and angular velocity are modeled by a random walk process, meaning $\mathbf{n}_{b_a} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\sigma}_{b_a}^2)$ and $\mathbf{n}_{b_\omega} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\sigma}_{b_\omega}^2)$, and the derivatives $\dot{\mathbf{b}}_{a_t} = \mathbf{n}_{b_a}$ and $\dot{\mathbf{b}}_{\omega_t} = \mathbf{n}_{b_\omega}$. For any two consecutive frames b_k and b_{k+1}, the pre-integration is given as

$$\begin{aligned} \boldsymbol{\alpha}_{b_{k+1}}^{b_k} &= \iint_{t\in[t_k,t_{k+1}]} \mathbf{R}_t^{b_k}(\hat{\mathbf{a}}_t - \mathbf{b}_{a_t})dt^2 \\ \boldsymbol{\beta}_{b_{k+1}}^{b_k} &= \int_{t\in[t_k,t_{k+1}]} \mathbf{R}_t^{b_k}(\hat{\mathbf{a}}_t - \mathbf{b}_{a_t})dt \\ \boldsymbol{\gamma}_{b_{k+1}}^{b_k} &= \int_{t\in[t_k,t_{k+1}]} \frac{1}{2}\boldsymbol{\gamma}_t^{b_k} \otimes \begin{bmatrix} 0 \\ \hat{\boldsymbol{\omega}}_t - \mathbf{b}_{\omega_t} \end{bmatrix} dt. \end{aligned} \tag{3}$$

The angular pre-integration $\boldsymbol{\gamma}_{b_{k+1}}^{b_k}$ is also considered as a rough rotation quaternion from b_{k+1} to b_k and is estimated from the angular velocity. However, given that the pre-integrations are only needed in the initialization process, and the previous value of the bias is not known, we can simplify the pre-integrations by setting the biases $\mathbf{b}_{a_t}$ and $\mathbf{b}_{\omega_t}$ to zero.

2) ORIENTATION ALIGNMENT

First, for $\mathbf{R}_c^b$, the following equation holds for any two consecutive image frames:

$$\mathbf{R}_{b_{k+1}}^{b_k} \mathbf{R}_c^b = \mathbf{R}_c^b \mathbf{R}_{c_{k+1}}^{c_k}, \tag{4}$$

where $\mathbf{R}_{b_{k+1}}^{b_k}$ can be calculated by the pre-integration $\boldsymbol{\gamma}_{b_{k+1}}^{b_k}$, or the attitude module. That is, $\mathbf{R}_{b_{k+1}}^{b_k} = \mathbf{R}_{b_k}^{w\,-1} \mathbf{R}_{b_{k+1}}^w$. With a quaternion representation, (4) can be rewritten as

$$\begin{aligned} \mathbf{q}_{b_{k+1}}^{b_k} \otimes \mathbf{q}_c^b &= \mathbf{q}_c^b \otimes \mathbf{q}_{c_{k+1}}^{c_k} \\ &\Leftrightarrow [\mathcal{L}(\mathbf{q}_{b_{k+1}}^{b_k}) - \mathcal{R}(\mathbf{q}_{c_{k+1}}^{c_k})]\mathbf{q}_c^b = \mathbf{0}, \end{aligned} \tag{5}$$

where the generation of quaternions is defined as $\mathbf{p} \otimes \mathbf{q} = \mathcal{L}(\mathbf{p})\mathbf{q} = \mathcal{R}(\mathbf{q})\mathbf{p}$.

In the initialization, a sliding window containing multiple frames is constructed, and $\mathbf{q}_c^b$ is solved by several homogeneous equations such as (5) using the Linear Least Squares (LLS) method. In addition, for $\mathbf{q}_l^w$, we have the following derivation based on (1):

$$\mathbf{q}_l^w \otimes \mathbf{q}_{c_k}^l = \mathbf{q}_{b_k}^w \otimes \mathbf{q}_c^b \Leftrightarrow \mathcal{R}(\mathbf{q}_{c_k}^l)\mathbf{q}_l^w = \mathbf{q}_{b_k}^w \otimes \mathbf{q}_c^b. \tag{6}$$

Similarly to $\mathbf{q}_c^b$, $\mathbf{q}_l^w$ can be solved by LLS.

3) LINEAR ALIGNMENT

After the orientation alignment, the status is shown in Fig. 2(b). Then, as shown in Fig. 2(c), linear alignment is performed to estimate the metric scale of the odometry. Thus, we define the states vector as

$$\mathcal{X}_I = \left[\mathbf{v}_{b_0}^{b_0\mathrm{T}}, \mathbf{v}_{b_1}^{b_1\mathrm{T}}, \ldots, \mathbf{v}_{b_n}^{b_n\mathrm{T}}, \mathbf{g}^{c_0\mathrm{T}}, s\right]^{\mathrm{T}}, \tag{7}$$

where $\mathbf{v}_{b_k}^{b_k}$ represents the velocity in the body frame in the kth image, b_0 and c_0 are the first frames of the body and the camera in the sliding window, respectively, $\mathbf{g}^{c_0}$ is represents gravity in the c_0 frame, and s is the metric scale of the odometry. For any consecutive frames b_k and b_{k+1} in the window, we have

$$\begin{aligned}
\boldsymbol{\alpha}_{b_{k+1}}^{b_k} &= \mathbf{R}_{c_0}^{b_k}(s(\mathbf{p}_{b_{k+1}}^{c_0} - \mathbf{p}_{b_k}^{c_0}) + \frac{1}{2}\mathbf{g}^{c_0}\Delta t_k^2 - \mathbf{R}_{b_k}^{c_0}\mathbf{v}_{b_k}^{b_k}\Delta t_k) \\
\boldsymbol{\beta}_{b_{k+1}}^{b_k} &= \mathbf{R}_{c_0}^{b_k}(\mathbf{R}_{b_{k+1}}^{c_0}\mathbf{v}_{b_{k+1}}^{b_{k+1}} + \mathbf{g}^{c_0}\Delta t_k - \mathbf{R}_{b_k}^{c_0}\mathbf{v}_{b_k}^{b_k}) \\
\mathbf{p}_{b_{k+1}}^{w} - \mathbf{p}_{b_k}^{w} &+ (\mathbf{R}_{b_{k+1}}^{w} - \mathbf{R}_{b_k}^{w})\mathbf{p}_c^b = s\mathbf{R}_l^w(\mathbf{p}_{c_{k+1}}^l - \mathbf{p}_{c_k}^l).
\end{aligned} \tag{8}$$

Combining with (1), we obtain the following linear measurement model:

$$\begin{aligned}
&\hat{\mathbf{z}}_{b_{k+1}}^{b_k} \\
&= \begin{bmatrix} \hat{\boldsymbol{\alpha}}_{b_{k+1}}^{b_k} - \mathbf{p}_c^b + \mathbf{R}_{c_0}^{b_k}\mathbf{R}_{b_{k+1}}^{c_0}\mathbf{p}_c^b \\ \hat{\boldsymbol{\beta}}_{b_{k+1}}^{b_k} \\ \mathbf{p}_{b_{k+1}}^{w} - \mathbf{p}_{b_k}^{w} + (\mathbf{R}_{b_{k+1}}^{w} - \mathbf{R}_{b_k}^{w})\mathbf{p}_c^b \end{bmatrix} = \mathbf{H}_{b_{k+1}}^{b_k}\mathcal{X}_I
\end{aligned} \tag{9}$$

$$\begin{aligned}
&\mathbf{H}_{b_{k+1}}^{b_k} \\
&= \begin{bmatrix} -\mathbf{I}\Delta t_k & \mathbf{0} & \frac{1}{2}\mathbf{R}_{c_0}^{b_k}\Delta t_k^2 & \mathbf{R}(\mathbf{p}_{c_{k+1}}^{c_0} - \mathbf{p}_{c_k}^{c_0}) \\ -\mathbf{I} & \mathbf{R}_{c_0}^{b_k}\mathbf{R}_{b_{k+1}}^{c_0} & \mathbf{R}_{c_0}^{b_k}\Delta t_k & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{R}_l^w(\mathbf{p}_{c_{k+1}}^l - \mathbf{p}_{c_k}^l) \end{bmatrix},
\end{aligned} \tag{10}$$

where $\mathbf{R}_{c_0}^{b_k}$, $\mathbf{R}_{b_{k+1}}^{c_0}$, $\mathbf{p}_{c_{k+1}}^{c_0}$, $\mathbf{p}_{c_k}^{c_0}$, $\mathbf{p}_{c_{k+1}}^{l}$, and $\mathbf{p}_{c_k}^{l}$ can be obtained from odometry when $\mathbf{R}_l^w$ and $\mathbf{R}_c^b$, which is estimated by the previous orientation alignment, are given. The variables $\mathbf{p}_{b_{k+1}}^{w}$, $\mathbf{p}_{b_k}^{w}$, $\mathbf{R}_{b_{k+1}}^{w}$, and $\mathbf{R}_{b_k}^{w}$ are obtained from attitude and GPS measurements, and Δt_k is the time interval between the kth and $k+1$th frames. Finally, the metric scale can be extracted by solving the following LLS problem:

$$\min_{\mathcal{X}_I} \sum_{k\in\mathcal{B}} \left\| \hat{\mathbf{z}}_{b_{k+1}}^{b_k} - \mathbf{H}_{b_{k+1}}^{b_k}\mathcal{X}_I \right\|^2. \tag{11}$$

B. SCALE INSENSITIVE POSE GRAPH OPTIMIZATION

An illustration of the scale insensitive pose graph optimization is shown in Fig. 3. The states, shown as gray circles, are the 6-DoF poses of all key frames after the initialization. The edge between two consecutive frames is the VO/VIO factor, and other items are global factors, including a barometer, GPS, and attitude sensor, among other factors. To decrease the computation time, we optimize the states when taking key frames instead of for every frame. Another reason is that optimization should not be done when the camera does not move. In Fig. 3, there are two parameters: M and N. The states

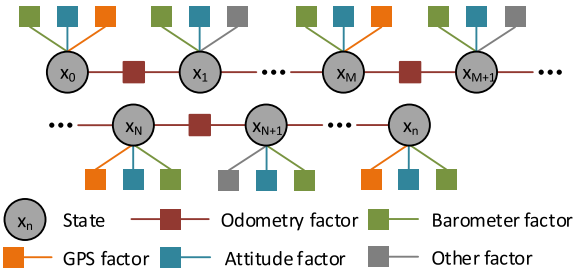

FIGURE 3. Illustration of the scale insensitive pose graph optimization. Each state represents the global 6-DoF pose of the current frame in the world frame, including position and orientation. The edge between two consecutive frames is the VO/VIO factor, and other items are global factors, including a barometer, GPS, and attitude and other factors potentially used.

before M are considered fully optimized states and are fixed in the following optimization to reduce the computation time. The states from M to n are optimized continuously to achieve a higher accuracy for those states. However, for the states from N to n, related factors are used to estimate the similarity transformation from the local frame to the global frame only, as drift normally occurs in a short movement. Thus, the old states contribute little to current drift.

1) GLOBAL POSE GRAPH OPTIMIZATION

The essence of optimization-based methods is the maximum likelihood estimation from the joint probability distribution of multiple sensor observations. The state vector is

$$\begin{aligned}
\mathcal{X}^w &= \left[\mathbf{x}_{b_0}^{w\mathrm{T}}, \mathbf{x}_{b_1}^{w\mathrm{T}}, \ldots \mathbf{x}_{b_M}^{w\mathrm{T}}, \ldots \mathbf{x}_{b_N}^{w\mathrm{T}}, \ldots \mathbf{x}_{b_n}^{w\mathrm{T}}, s, \theta_0\right]^{\mathrm{T}} \\
\mathbf{x}_{b_k}^{w} &= \left[\mathbf{p}_{b_k}^{w\mathrm{T}}, \mathbf{q}_{b_k}^{w\mathrm{T}}\right]^{\mathrm{T}}, k \in [0, n],
\end{aligned} \tag{12}$$

where $\mathbf{p}_{b_k}^w$ and $\mathbf{q}_{b_k}^w$ represent the position and quaternion of the body frame with respect to the global frame when taking the kth key frame, respectively, s is the scale of the latest $n-N+1$ states, and θ_0 is the yaw bias of attitude module, which is mainly caused by the magnetic declination. Suppose all measurements are independent of each other and any measurement error obeys a normal distribution: $\mathbf{p}(\mathbf{z}_k^j|\mathcal{X}^w) \sim \mathcal{N}(\hat{\mathbf{z}}_k^j, \boldsymbol{\Omega}_k^j)$. Given that the states before M are considered fixed, the optimization problem is converted to

$$\begin{aligned}
\mathcal{X}^{w*} &= \arg\max_{\mathcal{X}^w} \prod_{k=M}^{n}\prod_{j\in\mathcal{S}} p(\mathbf{z}_k^j|\mathcal{X}^w) \\
&= \arg\min_{\mathcal{X}^w} \sum_{k=M}^{n}\sum_{j\in\mathcal{S}} \left\| \mathbf{z}_k^j - \mathbf{h}_k^j\left(\mathcal{X}^w\right) \right\|_{\boldsymbol{\Omega}_k^j}^2,
\end{aligned} \tag{13}$$

where the Mahalanobis norm is defined as $\|\mathbf{r}\|_{\boldsymbol{\Omega}}^2 = \mathbf{r}^{\mathrm{T}}\boldsymbol{\Omega}^{-1}\mathbf{r}$, which places different weights on different measurements. The optimization problem has now been converted into a nonlinear least squares problem, which can be solved by the Gauss-Newton or Levenberg-Marquardt methods.

2) GLOBAL SENSOR FACTORS

First, we discuss the global factors. For the GPS factor, the original GPS measurement can be converted into a 3D position $\mathbf{p}_k^G$ in the ENU frame directly. The GPS factor is

$$\mathbf{z}_k^G - \mathbf{h}_k^G\left(\mathcal{X}^w\right) = \mathbf{p}_k^G - (\mathbf{p}_{b_k}^w + \boldsymbol{R}_{b_k}^w \mathbf{p}_G^b), \tag{14}$$

where $\mathbf{p}_G^b$ is the GPS position with respect to the body frame. The covariance is related to the number of satellites, which is normally given by the GPS receiver. Given that the GPS has a low frequency, we use a linear interpolation to double the frequency. There are also some cases in which the GPS is blocked by buildings, which leads to a very noisy measurement. Thus, to enhance the robustness in such cases, we establish two simple rules to decide whether to fuse GPS or not based on the covariance σ_k^{G2} and its derivative:

- Good to bad status: $(\sigma_k^G > 4 \&\& \dot{\sigma}_k^G > 0.2) || \sigma_k^G > 5$
- Bad to good status: $(\sigma_k^G > 5 \&\& \dot{\sigma}_k^G < -0.2) || \sigma_k^G < 4$

The thresholds are empirical values, and the values of specific thresholds may vary according to different types of GPS. The GPS factor is only fused when it is in a good status.

Barometer is used to measure air pressure, which can be converted to height by supposing the air pressure is constant at the same altitude in a period of time. Thus, the barometer factor is described as

$$\mathbf{z}_k^B - \mathbf{h}_k^B\left(\mathcal{X}^w\right) = p_k^B - [\mathbf{p}_{b_k}^w]_z, \tag{15}$$

where p_k^B is the height measurement and $[\mathbf{p}_{b_k}^w]_z$ represents the height of the up-to-optimized state.

The attitude is fused by the IMU and the magnetometer with high accuracy. However, the yaw angle of attitude has a bias due to the magnetic declination. Thus, the yaw bias is also involved during optimization. The attitude factor is derived as

$$\mathbf{z}_k^A - \mathbf{h}_k^A\left(\mathcal{X}^w\right) = 2\left[\mathbf{q}_k^A \otimes \mathbf{q}_{b_k}^{w\,-1} \otimes \begin{bmatrix}\cos(\theta_0/2) \\ \mathbf{r}\sin(\theta_0/2)\end{bmatrix}\right]_{xyz}, \tag{16}$$

where $\mathbf{q}_k^A$ is the attitude measurement, $\mathbf{r} = [0, 0, 1]^{\mathrm{T}}$, $[\mathbf{q}]_{xyz}$ is the imaginary part of the quaternion, and θ_0 is the yaw angle bias of the attitude module output.

3) LOCAL ODOMETRY FACTOR

As mentioned before, the odometry factor is used to constrain two consecutive frames due to its high accuracy estimation in a local region. The states between N to n are assumed to have the same metric scale. Thus, the odometry factor is

$$\begin{aligned}
&\mathbf{z}_k^L - \mathbf{h}_k^L\left(\mathcal{X}^w\right) \\
&\quad = \mathbf{z}_k^L - \mathbf{h}_k^L\left(\mathbf{x}_{b_k}^w, \mathbf{x}_{b_{k+1}}^w, s\right) \\
&\quad = \begin{bmatrix} s\mathbf{R}_c^{b\,-1}\mathbf{R}_{c_k}^{l\,-1}(\mathbf{p}_{c_{k+1}}^l - \mathbf{p}_{c_k}^l) - \mathbf{R}_{b_k}^{w\,-1}(\mathbf{p}_{b_{k+1}}^w - \mathbf{p}_{b_k}^w) \\ 2\left[\left(\mathbf{q}_{c_k}^{l\,-1} \otimes \mathbf{q}_{c_{k+1}}^l\right)^{-1} \otimes \left(\mathbf{q}_{b_k}^{w\,-1} \otimes \mathbf{q}_{b_{k+1}}^w\right)\right]_{xyz} \end{bmatrix}.
\end{aligned} \tag{17}$$

Once we obtain the metric scale of the odometry, a similarity transformation from the local frame of the odometry to the global frame is easily solved by the latest optimized global state and odometry state as follows:

$$\mathbf{S}_l^w = \mathbf{T}_{b_n}^w \mathbf{S}_c^b \mathbf{T}_{c_n}^{l\,-1}. \tag{18}$$

However, the estimation of $\mathbf{S}_l^w$ may be unstable because only one latest state is involved. Thus, a refinement of the $\mathbf{S}_l^w$ estimation is conducted by optimizing a sliding windows pose graph containing the latest $n - N + 1$ states. These states are considered with the same $\mathbf{S}_l^w$, and (18) is considered an initial value. The state vector and error vector of the refinement process are

$$\begin{aligned}
\mathcal{X}^l &= \left[\mathbf{q}_l^{w\mathrm{T}}, \mathbf{p}_l^{w\mathrm{T}}, s\right]^{\mathrm{T}} \\
\mathbf{e}_k(\mathcal{X}^l) &= \begin{bmatrix} \mathbf{p}_l^w - (\mathbf{p}_{b_k}^w + \mathbf{R}_{b_k}^w \mathbf{p}_c^b - s\mathbf{R}_{b_k}^w \mathbf{R}_c^b \mathbf{R}_{c_k}^{l\,-1}\mathbf{p}_{c_k}^l) \\ 2\left[\mathbf{q}_l^{w-1} \otimes \mathbf{q}_{b_k}^w \otimes \mathbf{q}_c^b \otimes \mathbf{q}_{c_k}^{l\,-1}\right]_{xyz} \end{bmatrix},
\end{aligned} \tag{19}$$

respectively. The optimal result $\mathcal{X}^{l*}$ can be obtained by solving the following nonlinear least squares problem

$$\mathcal{X}^{l*} = \arg\min_{\mathcal{X}^l} \sum_{k=N}^{n} \|\mathbf{e}_k(\mathcal{X}^l)\|_{\boldsymbol{\Omega}_k}. \tag{20}$$

The incoming odometry can be transformed directly into the global frame by the similarity transformation $\mathbf{S}_l^w$, which leads to an odometry-rate accurate global pose estimation.

C. VO/VIO FAULT DETECTION

In practical applications, the camera is easily interfered with by rapid rotation or a sudden change in illumination, which may lead to a faulty tracking. Here we show a simple odometry fault detection method based on a Chi-square test between odometry and global sensor measurements. The Chi-square distribution signifies the following: if n independent random variables all obey the standard normal distribution, a new random variable composed of the sum of the squares of these n random variables obeys the Chi-square distribution. By referring to the Chi-square distribution table and setting an appropriate Chi-square threshold, effective fault detection can be realized. Reference [26] utilized Chi-square test to detect the consistency problem in EKF-SLAM. In [27], a two state chi-square test was designed jointly with the fusion algorithm for fault detection using corrupted measurements. The following describes the use of GPS, barometer and attitude information to detect odometry faults.

GPS measurements at times k and $k+1$ are denoted as $\mathbf{p}_k^G$ and $\mathbf{p}_{k+1}^G$, respectively. Suppose their noise distributions obey the normal distribution. In addition, odometry outputs are represented by $\mathbf{p}_k^L$ and $\mathbf{p}_{k+1}^L$, and they obey normal distributions. These two types of measurements can both be converted into the global frame. The distance difference between these two converted measurements obey a normal distribution, which is defined as

$$\begin{aligned}
\Delta\mathbf{p} &= \left[\mathbf{p}_{k+m}^G - \mathbf{p}_k^G - (\mathbf{R}_{b_{k+m}}^w - \mathbf{R}_{b_k}^w)\mathbf{p}_G^b\right] \\
&\quad - \left[\mathbf{S}_l^w(\mathbf{p}_{c_{k+m}}^l - \mathbf{p}_{c_{k+m}}^l) - (\mathbf{R}_{b_{k+m}}^w - \mathbf{R}_{b_k}^w)\mathbf{p}_c^b\right] \\
&\quad \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\sigma}_p^2),
\end{aligned} \tag{21}$$

where $\boldsymbol{\sigma}_p^2$ can be determined by several related normal distributions. Given that GPS provides a 3D measurement, we have the following distribution:

$$\|\Delta\mathbf{p}\|^2_{\boldsymbol{\sigma}_p^2} = \Delta\mathbf{p}^{\mathrm{T}}{\boldsymbol{\sigma}_p^2}^{-1}\Delta\mathbf{p} \sim \chi^2(3). \tag{22}$$

For the barometer, only height is involved. Similarly to GPS, the difference norm in this cases obeys a $\chi^2(1)$ distribution. For attitude, the difference also obeys a $\chi^2(3)$ distribution, while the error is described by

$$\Delta\mathbf{q} = 2\left[\left(\mathbf{q}_c^b \otimes \mathbf{q}_{c_{k+m}}^l{}^{-1} \otimes \mathbf{q}_{c_k}^l \otimes \mathbf{q}_b^c\right)^{-1} \left(\mathbf{q}_{k+m}^A{}^{-1} \otimes \mathbf{q}_k^A\right)\right]_{xyz} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\sigma}_q^2). \tag{23}$$

Whenever the new measurements are incoming, a Chi-square test between odometry and other sensors can be performed. If the result is larger than a maximum threshold (such as the value that has a probability of 0.95) or smaller than a minimum threshold (such as the value that has a probability of 0.05), then the current odometry can be considered incorrect. However, in practice, one single test is not convincing enough to determine whether or not odometry is correct. Thus, consecutive Chi-square tests are required. When a fault is produced after consecutive tests for more than, for example, 80% of the times, then we can consider odometry to be incorrect. Once odometry becomes faulty, it needs to be restarted. During a restart, UAVs are normally still flying in uniform linear motion, which increases the difficulty in restarting the odometry, as mentioned before. After a successful restart, a scale insensitive initialization is conducted, as described in Section IV-A.

V. EXPERIMENTAL RESULTS

In this section, the experimental results are presented to verify the proposed SIMSF framework. We first introduce the overview of the experimental platform, sensors involved, and environment. Then we evaluate the accuracy, robustness, and real-time performance of our method without considering scale insensitivity. Finally, we give the scale insensitive results, including cases of using VO as input, scale drift, and in-air re-initialization.

A. EXPERIMENTAL SETUP

The UAV platform and all related sensors are shown in Fig. 4. The aircraft platform had a diagonal length of 550 mm and a total weight of 3.35 kg, including a 12000 mAh 6s LiPo battery. The GPS was a low-cost Ublox NEO-M8N, which has an absolute positioning accuracy of about 3 m in ideal conditions. The magnetometer was built into the GPS module to reduce electromagnetic interference from the UAV. An open source flight controller stack, Pixhawk, was used to control the UAV, which integrated a barometer and an IMU (MPU6050). The onboard processor was an NVIDIA Jetson TX2. Its processor is a combination of a dual core Denver2 and a four core ARM A57, with a 2-GHz processing frequency and an 8-GB memory. The performance of the

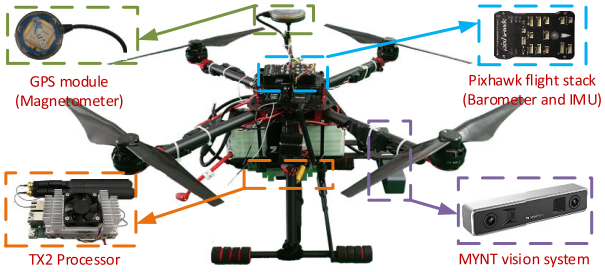

FIGURE 4. The UAV platform and all related sensors for the SIMSF experiments, including GPS, magnetometer, barometer, IMU, and vision camera.

TX2 processor was not very powerful, but the power consumption was only 7.5 W, and the weight was only 255 g, which is very suitable for payload-limited platforms such as UAVs. The vision module was a MYNT stereo camera, but only the left camera was used for fusion. To ensure that the image processing achieved real-time performance, the resolution of the image was set to 640 × 480 and the frequency was set to 10 Hz.

FIGURE 5. Experimental environment and flight trajectories (blue, green, and red lines) used to evaluate the accuracy, robustness, and real-time performance of the proposed methods.

The experimental environment and flight trajectories are shown in Fig. 5. We carried out three types of trajectory flight tests, which are shown as blue, green, and red lines in the figure. Trajectory I (blue lines) was a short-range flight. Trajectory II (green lines) was a medium-range flight that included a short indoor flight where there was no GPS signal, indicated by the yellow rectangle box in the figure. Trajectory III (red lines) was a long-range flight. In order to measure the accuracy of the algorithm, we used the results of loop closure [11] as the references. The accuracy, robustness, and real-time performance of the proposed methods are analyzed by the experimental results of these three flight trajectories.

B. OPTIMIZATION RESULTS

First, for Trajectory I, we compare the attitude constraint method in this paper with the magnetic constraint in VINS-Fusion [22]. The barometer constraint is not considered at this time. Fig. 6 shows the optimization trajectories of the proposed method, VINS-Fusion, VINS-Mono, and loop closure.

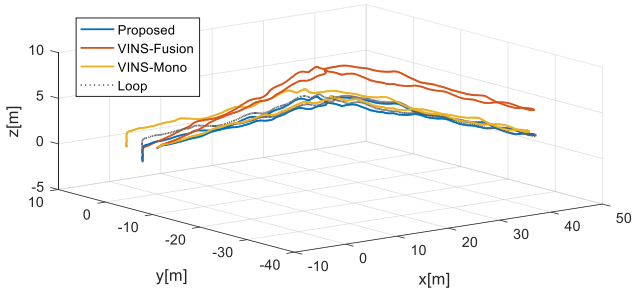

FIGURE 6. Flight trajectories recovered from VINS-Fusion, VINS-Mono, loop closure, and the proposed method.

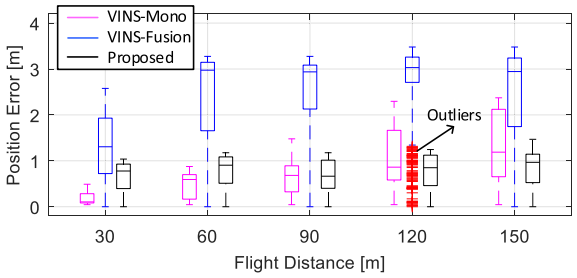

FIGURE 7. Position error box plots of VINS-Mono, VINS-Fusion, and the proposed method. Magnetic constraints are utilized in VINS-Fusion, while attitude constraints are used in our method.

The loop result is considered the reference trajectory. Fig. 7 shows the position error box plots of each method. Fig. 6 and Fig. 7 indicates that VINS-Mono drifted increasingly as the flight distance increased. Compared with the attitude constraint trajectory, the magnetic constraint trajectory has a large error in the vertical direction. Because magnetic constraints require accurate local magnetic field information, which is difficult to obtain in practice, the attitude constraint here is better than the magnetic constraint in VINS-Fusion. Furthermore, an incorrect magnetic constraint leads to the overall tilt of the optimized trajectory, as shown in Fig. 6. Therefore, in the following experiment, we use the attitude constraint instead of the magnetic constraint in VINS-Fusion.

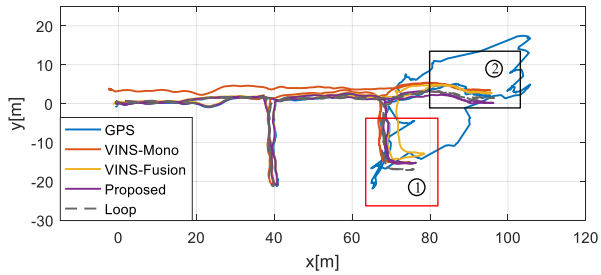

FIGURE 8. Flight trajectories recovered from GPS, VINS-Mono, VINS-Fusion, loop closure, and the proposed method with no GPS (area 1) or weak GPS signals (area 2).

Next, we analyze the optimization results for no or weak GPS signals in Trajectory II. Fig. 8 shows the trajectories of GPS, VINS-Mono, VINS-Fusion, loop closure, and our proposed method. The paths in the red rectangle (area 1) in the figure show that the UAV flew into the building. At this time, the GPS is inaccurate; its error is more than 10 meters, its measurement jumps are very large, and sometimes the GPS signal cuts out entirely. The paths in the black rectangle (area 2) show that the UAV flew close to the building, where the GPS is also blocked, but its errors and measurement jumps are smaller. Fig. 9 shows the error box plots over different time periods. The five corresponding time periods in the figure are: before entering area 1, flying in area 1, after leaving area 1 and before entering area 2, flying in area 2, and after leaving area 2. Due to the usage of visual information, in the case of the GPS signal being blocked (98 – 134s and 164 – 182s), an accurate position estimation can still be obtained for all methods. In the results of the last period (182 – 268s), there is a large drift for VINS-Mono, while the accuracy of other optimized results is higher due to fusing GPS information. Moreover, our result is more accurate than VINS-Fusion, especially when the GPS signal is weak. This is because when the GPS accuracy is lower than a certain threshold, we do not add a GPS constraint; we only use the barometer, vision, and attitude constraints to optimize. The above results show that our method has better robustness for conditions involving no or weak GPS signals.

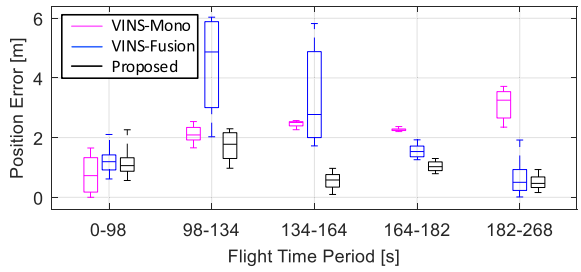

FIGURE 9. Position error box plots of VINS-Mono, VINS-Fusion, and the proposed method with no or weak GPS signals.

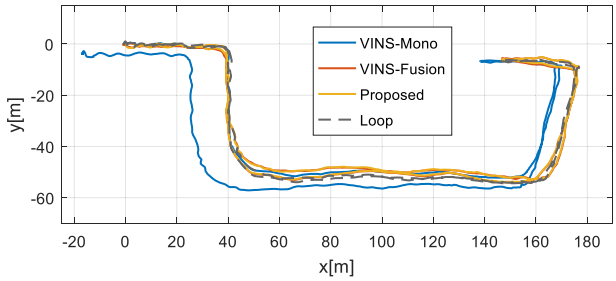

FIGURE 10. Flight trajectories recovered from VINS-Mono, VINS-Fusion, loop closure, and the proposed method for the long-distance flight of Trajectory III.

The real-time performance of the proposed method is analyzed according to the results of the long-distance flight Trajectory III. Fig. 10 shows the trajectories of VINS-Mono, VINS-Fusion, our proposed method, and loop closure. Fig. 11 shows the absolute position error box plot of each method.

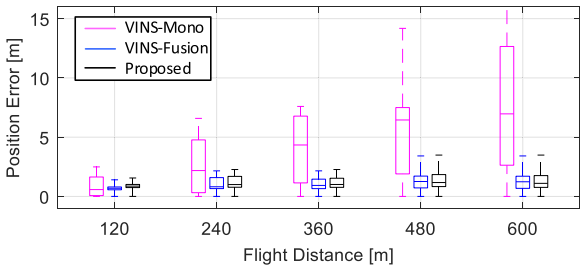

FIGURE 11. Position error box plots of VINS-Mono, VINS-Fusion, and the proposed method for the long-distance flight of Trajectory III.

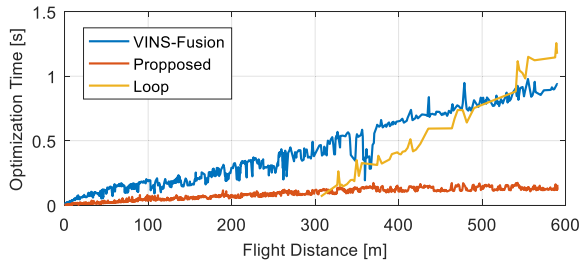

FIGURE 12. Optimization time of each method as a function of flight distance for the long-distance flight of Trajectory III.

These two figures suggest that the drift of VINS-Mono increases as the flight distance increases. After the optimization, VINS-Fusion and our proposed method can both obtain an accurate global position estimation without drift. Fig. 12 shows the optimization time of each method as the distance increases. The optimization time of our method is less than that of VINS-Fusion because we only add a constraint when a key frame is taken. Thus, the total number of optimization constraints is less than that of VINS-Fusion. As can be seen, the time required for loop optimization increases sharply as the scale increases. Therefore, the loop method is not suitable for large-scale scenes. Moreover, in order to ensure real-time performance, when the total number of states is larger than a certain threshold (1000 in the experiment), the states before this threshold are fixed without any optimization, so that the number of up-to-optimized states is limited. Thus, the optimization time of the proposed method is finally stable at about 0.15 s.

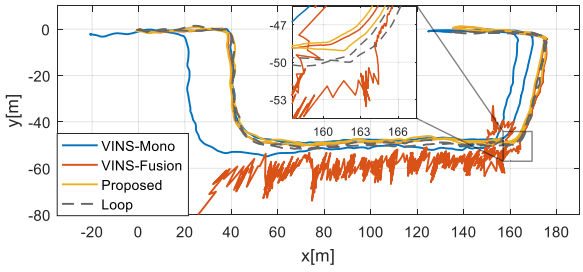

FIGURE 13. Real-time position estimations of VINS-Mono, VINS-Fusion, proposed method, and loop closure. For VINS-Fusion, once a latest position estimation is incorrect, the following estimations will be divergent.

Fig. 13 shows the real-time position estimations of VINS-Mono, VINS-Fusion, our proposed method, and loop closure, tested on another sequence same as Trajectory III. As can be seen in the zoomed-in area, for VINS-Fusion, once a position estimation is incorrect, the following estimations will be divergent. However, our method can provide good estimations for this sequence. This is because we use a number of the latest states to estimate the similarity transformation $\mathbf{S}_l^w$, while in VINS-Fusion, only one latest state is involved. Thus, we can achieve a better robustness against the outliers.

Table 1 shows the root mean square errors (RMSEs) of the different methods tested on various trajectories. The scale of VINS-Mono drifted by a very small extent (the errors are mainly caused by position drift), while our method also ignores the scale drift. From the table, we can conclude our method is more accurate than others in obtaining accurate global state estimations.

TABLE 1. RMSE [m] results of different algorithms for all three trajectories.

Trajectory	Num	Distance	RMSE [m]		
			VINS-Mono	VINS-Fusion	Proposed
I	01	158.6	**0.411**	0.605	0.589
	02	128.9	0.863	0.749	**0.701**
	03	157.5	0.703	0.711	**0.617**
	04	159.5	0.750	**0.515**	0.520
	05	146.6	1.495	1.332	**1.059**
II	01	293.9	1.900	2.079	**1.055**
	02	284.7	1.780	3.755	**0.801**
	03	301.2	1.724	3.546	**1.043**
	04	325.1	1.660	4.200	**1.432**
	05	294.0	2.621	2.703	**1.147**
III	01	579.4	8.874	1.958	**1.328**
	02	552.8	15.205	4.509	**2.388**
	03	607.4	12.984	1.382	**1.343**
	04	585.9	12.666	2.591	**2.150**
	05	585.1	14.096	2.065	**1.865**

C. SCALE INSENSITIVE RESULTS

To demonstrate the effectiveness and robustness of our method, we now present the scale insensitive results, including the cases of using VO as input, scale drift, and in-air re-initialization.

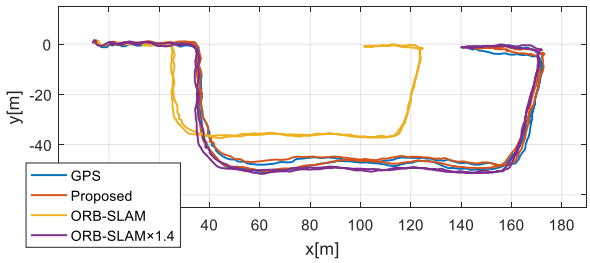

FIGURE 14. Trajectories recovered from GPS, ORB-SLAM, zoomed-in ORB-SLAM, and our proposed method.

First, the VO obtained from ORB-SLAM is considered the visual input of the system. As shown in Fig. 14, the

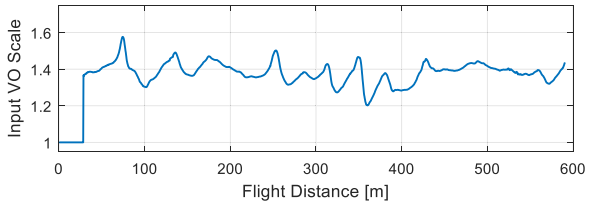

FIGURE 15. Scale estimation from scaled VO to the global frame.

trajectories are recovered from GPS, ORB-SLAM, and our method. The ORB-SLAM trajectory shown in the figure is scaled by our initialization process, and the initial scale is 68.5. However, the initial scale is not accurate. With the initial scale, we can transform VO to scaled VO, which means we can treat it as a VIO. Next, we estimate a new scale, namely the n-scale, which is the scale from the scaled VO to the global frame. Fig. 15 shows the n-scale estimation as the increase of the flight distance, suggesting an average value of about 1.4. This is also verified by Fig. 14: after ORB-SLAM is zoomed in 1.4 times, it is almost coincident with the GPS trajectory. Thus, we can conclude that our fusion framework has compatibility with VO input.

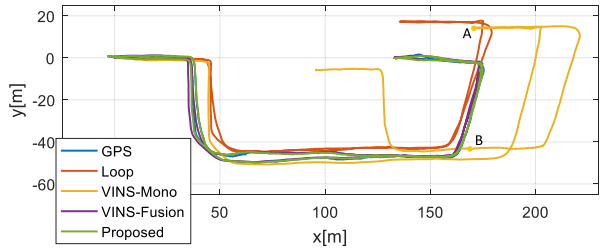

FIGURE 16. Trajectories recovered from GPS, loop closure, VINS-Mono, VINS-Fusion, and our proposed method. VINS-Mono drifted.

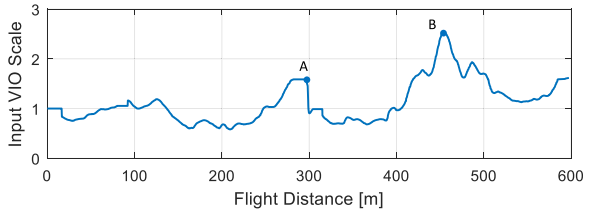

FIGURE 17. Scale estimation from the drifted VIO to the global frame.

We test our method on another trajectory, where the UAV is held by a person walking along the road. The recovered trajectories are shown in Fig. 16. Due to the vibration caused by walking, the VIO drifted almost all the time, which also lead to an incorrect loop closure. The n-scale from the VIO to the global frame is estimated in real-time and is shown in Fig. 17. There are two peak scales in Fig. 17 at points A and B, which are also shown in Fig. 16. A comparison of the trajectories of VINS-Mono and GPS shows that the movement of VINS-Mono is much smaller than that of GPS

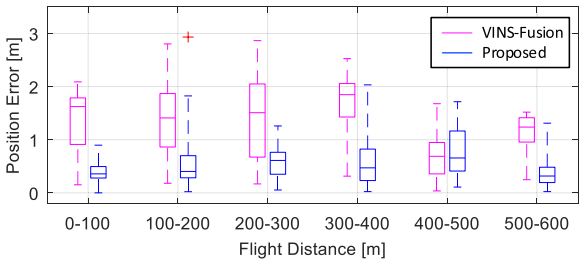

FIGURE 18. Position error box plots of VINS-Fusion and the proposed method as a function of flight distance in VIO drift conditions.

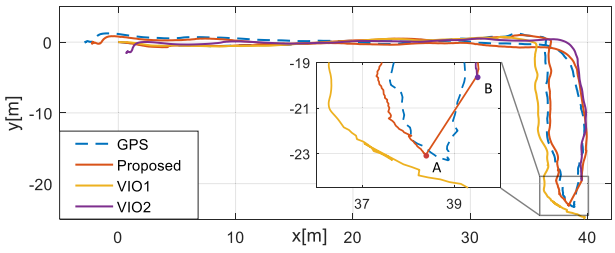

FIGURE 19. Trajectories recovered from GPS, our method, VINS-Mono (VIO1), and restarted VINS-Mono (VIO2).

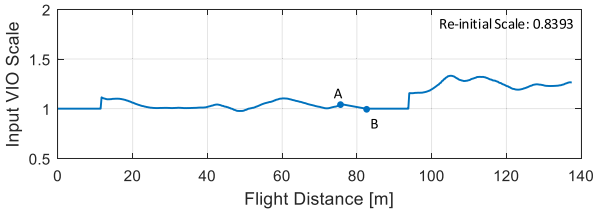

FIGURE 20. Scale estimation of VIO in the case of VIO restarting.

at points A and B. Thus, a bigger scale is needed to transfer VIO to the global frame (i.e., peaks A and B in Fig. 17). Given that Trajectory III was executed outdoors, GPS can provide a good position measurement. The position error box plots of VINS-Fusion and our proposed method, using the GPS trajectory as the reference, are shown in Fig. 18. The figure shows that our method can provide a higher accuracy estimation in scale drift conditions.

Finally, we choose another challenging case to verify our fault detection and in-air re-initialization methods. The trajectories recovered from GPS, our fusion method, VINS-Mono (VIO1), and restarted VINS-Mono (VIO2) after a fault are shown in Fig. 19. The fault of VIO is detected using our fault detection method at point A. Then, the VIO restarts and the re-initialization process provides an initial guess of the VIO scale, as in Section IV-A. The origin of the scaled VIO is aligned to the current GPS position at point B. Finally, the scale insensitive fusion is conducted as in Section IV-B. The scale estimation of the whole process is shown in Fig. 20, where A and B correspond to A and B in Fig. 19, respectively. After the VIO is restarted, an initial scale is estimated to be 0.84. Then the following n-scale is estimated based on the initial scale.

VI. CONCLUSIONS

In this paper, we proposed a scale insensitive multi-sensor fusion (SIMSF) framework for unmanned aerial vehicles based on graph optimization. Our method can fuse local measurements of VO/VIO and global measurements of GPS, attitude sensors, and barometer to achieve locally accurate and globally drift-free state estimations. We utilized certain strategies, such as attitude constraints, GPS status evaluation, and key frame-based optimization to enhance the accuracy, robustness, and real-time performance of our method. In addition, multiple latest states were used to increase the stability of the optimization process against outliers.

Furthermore, by estimating the initial scale in the initialization, the fusion framework is compatible with any unscaled odometry. In addition, by estimating the scale of the VIO or scaled VO in real-time graph optimization, the fusion framework can still achieve accurate estimations in the case of scale drifting. Finally, by adding a fault detection module and re-initialization strategy, the system can recover from the fault status rapidly and provide accurate state estimations again. A comparison between our results and the results from other state-of-the-art methods employed in various challenging conditions demonstrate the superior accuracy, real-time performance, and robustness of our fusion framework.

REFERENCES

[1] T. Tomic, K. Schmid, P. Lutz, A. Domel, M. Kassecker, E. Mair, I. Grixa, F. Ruess, M. Suppa, and D. Burschka, "Toward a fully autonomous UAV: Research platform for indoor and outdoor urban search and rescue," *IEEE Robot. Autom. Mag.*, vol. 19, no. 3, pp. 46–56, Sep. 2012.

[2] J. A. Gonçalves and R. Henriques, "UAV photogrammetry for topographic monitoring of coastal areas," *ISPRS J. Photogramm. Remote Sens.*, vol. 104, pp. 101–111, Jun. 2015.

[3] Z. Wang, Y. Wu, and Q. Niu, "Multi-sensor fusion in automated driving: A survey," *IEEE Access*, vol. 8, pp. 2847–2868, 2020.

[4] J. Zhang and S. Singh, "Loam: Lidar odometry and mapping in real-time," *Robot., Sci. Syst.*, vol. 2, no. 9, pp. 1–9, 2014.

[5] T. Shan and B. Englot, "LeGO-LOAM: Lightweight and ground-optimized lidar odometry and mapping on variable terrain," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2018, pp. 4758–4765.

[6] D. Scaramuzza and F. Fraundorfer, "Visual odometry [tutorial]," *IEEE Robot. Autom. Mag.*, vol. 18, no. 4, pp. 80–92, Dec. 2011.

[7] J. Engel, J. Sturm, and D. Cremers, "Semi-dense visual odometry for a monocular camera," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 1449–1456.

[8] Y. Lu and D. Song, "Robust RGB-D odometry using point and line features," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 3934–3942.

[9] S. Leutenegger, S. Lynen, M. Bosse, R. Siegwart, and P. Furgale, "Keyframe-based visual–inertial odometry using nonlinear optimization," *Int. J. Robot. Res.*, vol. 34, no. 3, pp. 314–334, Mar. 2015.

[10] M. Bloesch, S. Omari, M. Hutter, and R. Siegwart, "Robust visual inertial odometry using a direct EKF-based approach," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2015, pp. 298–304.

[11] T. Qin, P. Li, and S. Shen, "VINS-mono: A robust and versatile monocular visual-inertial state estimator," *IEEE Trans. Robot.*, vol. 34, no. 4, pp. 1004–1020, Aug. 2018.

[12] A. Angeli, S. Doncieux, J.-A. Meyer, and D. Filliat, "Real-time visual loop-closure detection," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2008, pp. 1842–1847.

[13] R. Mur-Artal and J. D. Tardos, "ORB-SLAM2: An open-source SLAM system for monocular, stereo, and RGB-D cameras," *IEEE Trans. Robot.*, vol. 33, no. 5, pp. 1255–1262, Oct. 2017.

[14] S. Lynen, M. W. Achtelik, S. Weiss, M. Chli, and R. Siegwart, "A robust and modular multi-sensor fusion approach applied to MAV navigation," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Nov. 2013, pp. 3923–3929.

[15] S. Weiss, M. W. Achtelik, M. Chli, and R. Siegwart, "Versatile distributed pose estimation and sensor self-calibration for an autonomous MAV," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2012, pp. 31–38.

[16] S. Shen, Y. Mulgaonkar, N. Michael, and V. Kumar, "Multi-sensor fusion for robust autonomous flight in indoor and outdoor environments with a rotorcraft MAV," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2014, pp. 4974–4981.

[17] J. Rehder, K. Gupta, S. Nuske, and S. Singh, "Global pose estimation with limited GPS and long range visual odometry," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2012, pp. 627–633.

[18] C. Merfels and C. Stachniss, "Pose fusion with chain pose graphs for automated driving," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2016, pp. 3116–3123.

[19] H. Oleynikova, M. Burri, S. Lynen, and R. Siegwart, "Real-time visual-inertial localization for aerial and ground robots," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2015, pp. 3079–3085.

[20] R. Mascaro, L. Teixeira, T. Hinzmann, R. Siegwart, and M. Chli, "GOMSF: Graph-optimization based multi-sensor fusion for robust UAV pose estimation," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2018, pp. 1421–1428.

[21] J. Surber, L. Teixeira, and M. Chli, "Robust visual-inertial localization with weak GPS priors for repetitive UAV flights," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2017, pp. 6300–6306.

[22] T. Qin, S. Cao, J. Pan, and S. Shen, "A general optimization-based framework for global pose estimation with multiple sensors," 2019, *arXiv:1901.03642*. [Online]. Available: http://arxiv.org/abs/1901.03642

[23] H. Strasdat, J. M. M. Montiel, and A. J. Davison, "Visual SLAM: Why filter?" *Image Vis. Comput.*, vol. 30, no. 2, pp. 65–77, Feb. 2012.

[24] R. Mahony, T. Hamel, and J.-M. Pflimlin, "Complementary filter design on the special orthogonal group SO(3)," in *Proc. 44th IEEE Conf. Decis. Control*, Dec. 2005, pp. 1477–1484.

[25] K. J. Wu, C. X. Guo, G. Georgiou, and S. I. Roumeliotis, "VINS on wheels," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2017, pp. 5155–5162.

[26] T. Bailey, J. Nieto, J. Guivant, M. Stevens, and E. Nebot, "Consistency of the EKF-SLAM algorithm," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Oct. 2006, pp. 3562–3568.

[27] C. Yang, A. Mohammadi, and Q.-W. Chen, "Multi-sensor fusion with interaction multiple model and chi-square test tolerant filter," *Sensors*, vol. 16, no. 11, p. 1835, Nov. 2016.

BO DAI received the B.Eng. degree in engineering and automation from the University of Science and Technology Beijing, Beijing, China, in 2014. He is currently pursuing the Ph.D. degree in mechatronics engineering with the Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang, China, and the University of the Chinese Academy of Sciences, Beijing.

His main research interests include UAV control, visual SLAM, and multi-sensor fusion.

YUQING HE (Member, IEEE) received the B.S. degree in engineering and automation from Northeastern University, Qinhuangdao, China, in 2002, and the Ph.D. degree in pattern recognition and intelligent systems from the Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang, China, in 2008.

In 2012, he was a Visiting Researcher at the Institute for Automatic Control Theory, Technical University of Dresden, Germany. He is currently a Professor with the State Key Laboratory of Robotics, Shenyang Institute of Automation, Chinese Academy of Sciences. His main research interests include nonlinear estimation, control, and the cooperation of multiple robots.

LIYING YANG (Member, IEEE) received the B.S. and M.S. degrees in automatic control engineering from Shenyang Jianzhu University, Shenyang, China, in 2002 and 2005, respectively, and the Ph.D. degree in pattern recognition and intelligent systems from the Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang, in 2011.

She is currently an Associate Professor with the State Key Laboratory of Robotics, Shenyang Institute of Automation, Chinese Academy of Sciences. Her current research interests include control and the path planning of aerial vehicles.

YUN SU received the B.Eng. degree from the Department of Automation, Chang'an University, China, in 2015. He is currently pursuing the Ph.D. degree with the Shenyang Institute of Automation, Chinese Academy of Sciences, and the University of the Chinese Academy of Sciences, Beijing, China.

His main research interests include robot collaboration control, robot localization, mapping algorithms, and visual SLAM.

YUFENG YUE (Member, IEEE) received the B.Eng. degree in automation from the Beijing Institute of Technology, Beijing, China, in 2014, and the Ph.D. degree from Nanyang Technological University, Singapore, in 2019.

He was a Visiting Scholar with the University of California at Los Angeles, Los Angeles, CA, USA, in 2019. He is currently a Research Fellow of the Advanced Robotics Lab, Nanyang Technological University. His research interests include mapping, navigation coordination, and reasoning for multi-robot systems in complex environments.

WEILIANG XU (Senior Member, IEEE) received the B.E. degree in manufacturing engineering and the M.E. degree in mechanical engineering from Southeast University, Nanjing, China, in 1982 and 1985, respectively, and the Ph.D. degree in mechatronics and robotics from Beihang University, Beijing, China, in 1988.

He joined The University of Auckland, Auckland, New Zealand, in February 1, 2011, as the Chair of mechatronics engineering. His current research interests include advanced mechatronics and robotics with applications in medicine and food.