Multimodal Information Fusion for High-Robustness and Low-Drift State Estimation of UGVs in Diverse Scenes

Dongjie Wu Xunyu Zhong Xiafu Peng Huosheng Hu *Life Senior Member, IEEE*, and Qiang Liu

***Abstract*— Currently, the autonomous positioning of unmanned ground vehicles (UGVs) still faces the problems of insufficient persistence and poor reliability, especially in the challenging scenarios where satellites are denied, or the sensing modalities such as vision or laser are degraded. Based on multimodal information fusion and failure detection (FD), this article proposes a high-robustness and low-drift state estimation system suitable for multiple scenes, which integrates light detection and ranging (LiDAR), inertial measurement units (IMUs), stereo camera, encoders, attitude and heading reference system (AHRS) in a loose coupling way. Firstly, a state estimator with variable fusion mode is designed based on the error-state extended Kalman filtering (ES-EKF), which can fuse encoder-AHRS subsystem (EAS), visual-inertial subsystem (VIS), and LiDAR subsystem (LS) and change its integration structure online by selecting a fusion mode. Secondly, in order to improve the robustness of the whole system in challenging environments, an information manager is created, which judges the health status of subsystems by degeneration metrics, and then online selects appropriate information sources and variables to enter the estimator according to their health status. Finally, the proposed system is extensively evaluated using the datasets collected from six typical scenes: street, field, forest, forest-at-night, street-at-night and tunnel-at-night. The experimental results show our framework is better or comparable accuracy and robustness than existing publicly available systems.**

***Index Terms*— Error-state extended Kalman filter (ES-EKF), failure detection (FD) and handling, light detection and ranging (LiDAR)-inertial-visual-encoder odometry, multimodal information fusion, state estimation.**

I. Introduction

UNMANNED ground vehicles (UGVs) have been widely deployed in various real world applications, such as demining robots [1], transportation robots [2], reconnaissance unmanned vehicles [3], etc. However, with the expansion of motion range and the increase of task types, UGVs often encounter more challenging environments that often lead to significant degradation of certain sensing modalities, resulting in the deterioration of the reliability and accuracy of the positioning system.

The global navigation satellite system (GNSS) has been widely deployed for the positioning of UGVs, which however is extremely vulnerable to the blockage of high buildings and malicious interference [4], [5] (such places are called satellite-denied areas). The navigation system based on MEMS-inertial measurement unit (IMU) or encoder is cheap and simple, but the positioning errors accumulate rapidly with time and distance [6], [7]. Recently, many state estimation algorithms using camera or light detection and ranging (LiDAR) have emerged, which can estimate pose with low drift over a long range. However, when encountering degenerate scenes, such as lack of texture [6], [8] for camera and scarcity of geometrical structure for LiDAR [9], [10], these methods often degenerate seriously or even fail. The systems based on a single sensing modality often cannot meet the needs of reliable positioning in complex environment. Fortunately, different modalities are complementary to each other, so multimodal fusion technology can be used to improve the performance of the state estimation.

On the other hand, a large number of studies have shown that, in a multimodal fusion system, one of the ways to improve its resilience is to effectively detect and deal with abnormal conditions, especially the failure problems caused by environmental modal degradation. This requires giving the whole architecture the abilities: can find the failed modality according to certain metrics; can isolate it from the fusion center to prevent it from deteriorating the whole system; can add it to the fusion algorithm when the modality returns to normal. Up to now, there are mainly three methods to handle the problem of degeneracy: 1) switching to another system [9]; 2) predicting the state using a constant velocity model or history data [11]; and 3) only predicting state in degenerate directions [12]. However, none of them are satisfactory as the first method requires a backup system that is not used most of the time, and the latter two methods cannot deal with long-term degradation.

Therefore, from the perspective of real-world application, this article presents a high-robustness and low-drift state estimation system suitable for multiple scenarios. The test

Manuscript received 27 April 2022; revised 18 July 2022; accepted 30 August 2022. Date of publication 13 September 2022; date of current version 23 September 2022. This work was supported in part by the Equipment Advance Research Funds under Grant 61405180205, in part by the National Natural Science Foundation of China under Grant 61703356, in part by the Fundamental Research Funds for the Central Universities under Grant 20720190129, and in part by the Aeronautical Science Foundation of China under Grant 201908068003. The Associate Editor coordinating the review process was Dr. Sougata Kumar Kar. *(Corresponding author: Xunyu Zhong.)*

Dongjie Wu, Xunyu Zhong, and Xiafu Peng are with the Department of Automation, School of Aerospace Engineering, Xiamen University, Xiamen 361102, China (e-mail: wudongjiechn@outlook.com; zhongxunyu@xmu.edu.cn; xfpeng@xmu.edu.cn).

Huosheng Hu is with the School of Computer Science and Electronic Engineering, University of Essex, CO4 3SQ Colchester, U.K. (e-mail: hhu@essex.ac.uk).

Qiang Liu is with the Department of Psychiatry, University of Oxford, OX3 7JX Oxford, U.K. (e-mail: qiang.liu@psych.ox.ac.uk).

Digital Object Identifier 10.1109/TIM.2022.3205687

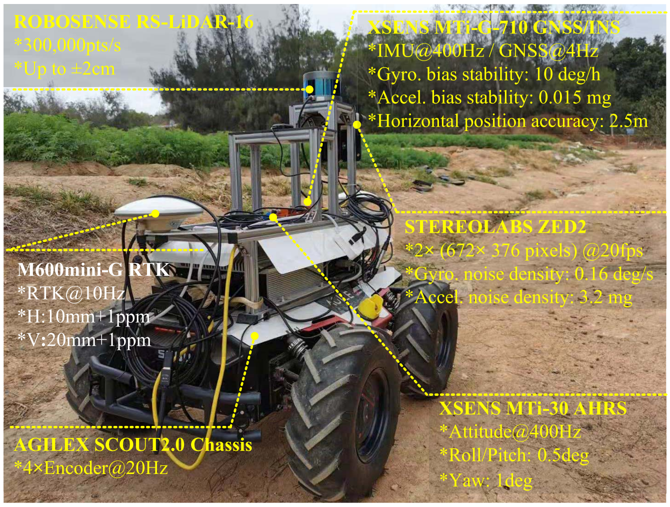

Fig. 1. Our UGV, AGILEX SCOUT2.0, is equipped with a XSENS MTi-G-710 for inertial data, a XSENS MTi-30 AHRS for 3-D attitude, a STEREOLABS ZED2 for stereo camera images and inertial data, a ROBOSENSE RS-LiDAR-16 for point cloud data, four encoders integrated into the vehicle for motor speed data. M600mini-G real time kinematic (RTK) system and XSENS MTi-G-710 GNSS/INS system do not participate in the multimodal fusion, and they are only used for providing position reference abbreviated as RTK data and GNSS data, respectively. The overall dimensions of the testing vehicle are about 940 × 700 × 910 mm (length × width × height).

platform and its sensor configuration are shown in Fig. 1, and the system overview is shown in Fig. 2. It can effectively detect and handle the degradation of vision or laser modality, and continuously provide reliable and real-time state information for UGVs. To summarize, the main innovations and contributions of this article are as follows.

1) A low drift, highly robust state estimation system suitable for multiple typical scenarios, which can continuously and reliably provide pose estimate even in extremely challenging environments, such as vision or laser significant degradation scenes. To the best of our knowledge, this is the first system fusing 3-D LiDAR, IMUs, stereo camera, encoders, and attitude and heading reference system (AHRS) in the open literature (see Section III).
2) An error-state extended Kalman filtering (ES-EKF)-based state estimator with four fusion modes is designed to fuse encoder-AHRS subsystem (EAS), visual-inertial subsystem (VIS), and LiDAR subsystem (LS), in which each fusion mode represents a way that subsystems participate in fusion. And its fusion mode can be changed online according to the health status of the subsystems (see Section V).
3) A novel failure detecting and handling strategy is proposed, which independently evaluates the health status of VIS and LS, and then with the cooperation of the variable fusion mode estimator, dynamically selects appropriate information sources and their variables to participate in the fusion (see Section VI).
4) A comprehensive system evaluation was completed on the datasets of six scenes, i.e., street, field, forest, forest-at-night, street-at-night, and tunnel-at-night. As far as we know, the scenes given in this article are the most abundant and comprehensive in the research of multimodal fusion state estimation (see Section VII).

The rest of the article is organized as follows. Section II outlines some previous work related to multisensor fusion for the state estimation of UGVs in challenging environments. In Section III, the proposed system architecture is explained. The workflow of the three subsystems, EAS, VIS and LS is introduced in Section IV. Section V describes the proposed algorithm for data fusion, namely Variable Fusion Mode State Estimator. Information manager is described in Section VI, including timestamp alignment and failure handling. Experimental results are given in Section VII to demonstrate the feasibility and performance of the proposed system. Finally, a brief conclusion and future work are presented in Section VIII.

II. Related Work

A. State Estimation Algorithm

In the field of robotics, state estimation algorithms mainly include filtering-based method, such as extended Kalman filtering (EKF) [5], [13] and unscented Kalman filtering (UKF) [14], [15], and optimization-based method, such as sliding-window graph optimization [10], [16] batch graph optimization [17]. Generally, the former is suitable for applications where lower computational cost is desired, and the latter is suitable for computationally intensive tasks such as smoothing and mapping.

In order to compromise the computational cost and the performance of state estimation, in our system, a sliding-window graph optimization method is used in the VIS and LS, and the ES-EKF method is used in the final fusion stage. It is worth mentioning that ES-EKF is a novel filtering algorithm designed on 3-D rotation Lie groups, which was first used in the field of aircraft [18]. From the perspective of the filter structure, the traditional EKF directly estimates the state variable itself (often called the full state), and ES-EKF estimates the error between the true state and its estimate (often called error state). The latter has some obvious advantages over the former [19]: 1) the attitude error can be expressed as a 3-D vector; 2) error state is small, which can reduce the nonlinear complexity of the system model; and 3) error state changes slowly, which is beneficial to reduce the update frequency.

B. LiDAR-Inertial-Visual System

There is not much work on odometry using LiDAR, IMU, and camera at the same time. In the early stage, its applications were mainly limited by computing power, but in the recent ten years, with the significant increase in computing power, it has been becoming a hot research direction of unmanned vehicle resilient navigation. From the perspective of system architecture, existing multimodal fusion odometry systems can be mainly divided into three categories: 1) centralized; 2) distributed-cascade; and 3) distributed-parallel.

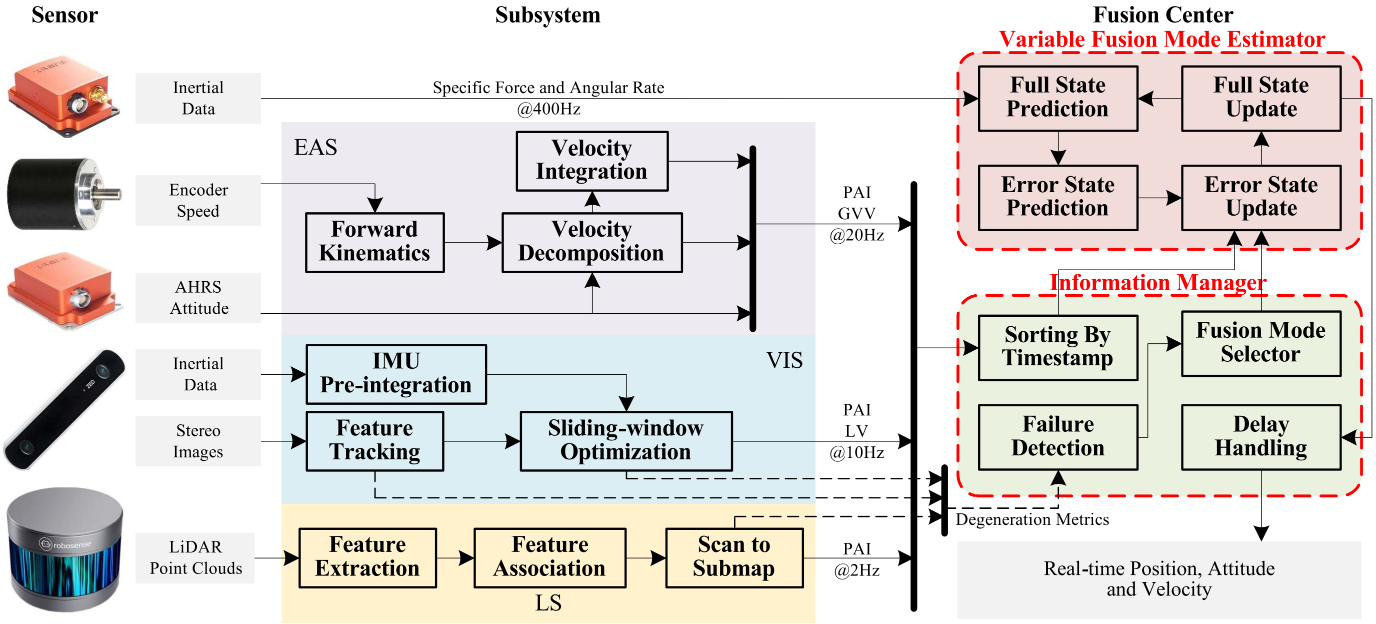

Fig. 2. System overview.

In a centralized system, raw data from multiple sensors is compressively processed to obtain certain features, such as corner and line features extracted from images, edge, and surface points extracted from point clouds, and then these features are sent to the fusion center to estimate the pose of a vehicle. For example, Zuo *et al.* [20] present a multi-state constraint Kalman filter (MSCKF)-based LiDAR-inertial-camera odometry, called LIC-Fusion, which fuses inertial data, extracted LiDAR points, and sparse visual features. The average of average absolute trajectory error (ATE) is about 4.06 m on the outdoor dataset around 800 m in length. Soon after, LIC-Fusing 2.0 [21] is proposed on the basis of LIC-Fusion, which introduces a sliding-window plane-feature tracking for high-quality data association to make point cloud matching more robust. Lin and Zhang [23] successively propose two systems, termed R^{2}LIVE [22] and R^{3}LIVE, based on error-state iterated Kalman filter (ESIKF) and composed of solid-state LiDAR, IMU and monocular camera. Evaluations for the two systems validate that they are able to run in challenging scenario, even in narrow tunnel-like environments. Although the codes of R^{2}LIVE and R^{3}LIVE have been open source, they are currently only applicable to solid-state LiDAR, and their code has not yet been adapted to mechanical LiDAR.

In a distributed-cascade system, there are multiple subsystems, and state estimates output by one subsystem will be used as the input of another subsystem, which will be carried out in turn to refine the system's state estimation. For example, Zhang and Singh [24] proposed a sequential-processing system, in which the IMU mechanization provides a rough prediction of motion at high frequency, then the visual-inertial odometry estimates the system state at an intermediate frequency, and finally, the subsystem for point cloud matching further refines the motion estimation at low frequency. Li [9] designed an optimization-based simultaneous localization and mapping (SLAM) system consisting of LiDAR odometry, monocular visual-inertial odometry, odometry selection module, and mapping module. Among these modules, the selection module is used to select one appropriate odometry source for mapping, so as to alleviate the possible accuracy drop caused by visual or laser modality degradation. The overall mean translation error relative to distance traveled is about 0.81% evaluated on the KITTI odometry dataset [25], [26]. Palieri *et al.* [27] propose a optimization-based LiDAR-centric robust odometry solution, LOCUS, in which a health monitor module is created to select the most appropriate one from subsystems (e.g., AHRS and wheel-inertial odometry) as the initial guess of subsequent point cloud registration.

A distributed-parallel system are usually composed of multiple subsystems and a fusion center, and the state estimates output by each subsystem are sent to the fusion center for further fusion at the pose or velocity level. For example, Kubelka *et al.* [28] designed an EKF-based positioning system for skid-steer vehicle for urban search and rescue missions, which fuses LiDAR odometry, IMU odometry, omnidirectional visual odometry and track odometry. Their system's overall median accuracy of about 1.2% and 1.4% of the total distance traveled indoor and outdoor, respectively. In order to achieve robust solution to general localization in challenging scenes, Simanek *et al.* [29] added an anomaly detection module to the same system, which can identify and reject the actual abnormal data. Shan *et al.* [12] design a factor graph-based LiDAR-visual-inertial state estimation framework, LVI-SAM, whose subsystems can work independently when failure is detected in one of them, or jointly when no failure occurs.

It is evident from the system overview in Fig. 2 that our proposed system is also a distributed-parallel. From the point of view of system reliability, distributed-parallel systems have the following advantages: 1) each subsystem is independent of each other and has high independence, which facilitates the isolation and addition of information sources and 2) a subsystem is usually constructed based on one modality, so an efficient failure detection (FD) method can be designed according to the characteristics of a single mode, and a reliable decision-making basis for failure handling can be provided.

C. FD and Handling

Studies have shown that vision and laser systems will degenerate in the absence of texture and geometric features, respectively, and such environmental degeneracy often leads to failure of estimation task [30]. Therefore, the FD and handling of subsystems or modules is an indispensable part to improve system reliability in practical applications of navigation systems. At present, there is no unified FD method, and most of the existing methods are specially designed for a certain modality because different modalities have different characteristics.

Simanek *et al.* [29] trained a ternary Gaussian mixture model to detect anomalies for the EKF fusion algorithm so that the state estimator has good performance in multiple scenarios. But, the disadvantage of the data-based method is that it cannot guarantee that the classifier has good generalization ability for unknown environments. Zhang *et al.* [30] proposed a failure discrimination method based on eigenvalue analysis for the optimization-based state estimation system, and this method has been applied in many open-source laser odometry systems, such as LeGO-LOAM [31] and LIO-SAM [32]. However, this eigenvalue analysis method is usually difficult to apply to visual or visual-inertial odometry systems with a lot of variables to be optimized. For systems based on visual modality, the common way to judge whether the system is normal is to check whether the output variables, such as the number of tracked features and the estimated IMU bias, are normal, and this output variables-based method is used in many classic visual positioning systems, e.g., visual-inertial navigation system (VINS)-Mono [16], VINS-Fusion [33]. In LOCUS, a simple rate-check: if subsystem messages are at a sufficient rate (>1 Hz), then the source is healthy, is chosen as a health metric, and obviously, this health metric can only indicate whether there is data, but cannot measure the quality of the data.

Once a failure event is detected, it needs to be dealt with to reduce the impact on the positioning performance of the whole system as much as possible. A common failure handling method is to isolate the failed subsystem source from the processing pipeline, and then add it to participate in the fusion once it returns to normal. For example, Zhang and Singh [24] design an automatic reconfiguration strategy for bypassing the failed subsystem to handle vision or laser degeneracy in their distributed-cascade system. In the system designed by Li [9], a subsystem selection module can select a subsystem without failure to provide the initial guess value for the next-level laser subsystem. A health monitor module is designed in the LOCUS [27], which plays a very similar function to the selection module in the system of Li.

In our system, the eigenvalue analysis-based and the output variables-based method are applied to detect the failure of LS and VIS, respectively. Compared with the existing failure processing strategies, the strategy proposed in this article first selects the available subsystems according to the health status of the subsystems, and then selects complementary variables from their output variables to participate in the fusion according to their characteristics, instead of sending all output variables to the estimator. This can make full use of the complementarity between different modalities and further improve the performance of the system. For example, for VIS, when LS is normal, its velocity estimation is fused, and its pose increment estimation is fused otherwise.

III. System Overview and Attitude Representation

A. System Overview

Fig. 2 shows the system overview of our proposed fusion framework, which consists of two parts: *subsystem* part and *fusion center* part. The three subsystems are: EAS, VIS, and LS. The EAS receives input from four encoders and an AHRS, then outputs position and attitude increment (PAI) and global vertical velocity (GVV) measurements. The VIS fuses stereo images and inertial data to produce PAI and local velocity (LV) measurements based on sliding-window factor graph optimization [33]. The LS processes LiDAR point cloud data to obtain the PAI measurements using LOAM-based matching algorithm [34].

The *variable fusion mode estimator* is a multimodal information fusion algorithm based on ES-EKF: 1) firstly, it uses IMU-driven kinematics to predict the full state of the system; 2) secondly, it uses the Kalman filter to directly estimate the error of the system; and 3) finally, it composes the error estimation with the predicted value of the full state to obtain the full state estimation. The *information manager* is mainly responsible for detecting and handling the subsystem failures caused by the degeneracy problem of laser or vision modalities and configuring an appropriate fusion mode for the estimator according to the FD results, improving the robustness of the whole system in a challenging environment. Among them, the *degeneration metrics* are variables from VIS and LS, which can online reflect whether they fail. See below for a detailed description of each module in the figure.

B. Attitude Representation

The global coordinate frame is denoted as $\mathcal{G}$, in which z-axis is opposite to gravity, and x–z-axes conform to the right-hand relationship, e.g., east-north-up frame. Local coordinate frame are denoted as $\mathcal{L}$, which is fixedly connected to the robot, e.g., chassis frame. Let $\mathbb{R}^3$ and SO(3) represent a 3-D Euclidean space and a 3-D rotation group, respectively. For a certain space attitude of $\mathcal{L}$ with respect to $\mathcal{G}$, there are various forms of parametric representation, such as rotation matrix $\boldsymbol{R} \in \mathrm{SO}(3)$, rotation vectors $\boldsymbol{\phi} \in \mathbb{R}^3$ and Euler angles $\boldsymbol{\theta} \in \mathbb{R}^3$, etc. According to [19], $\boldsymbol{R}$ and $\boldsymbol{\phi}$ satisfy the following conversion relationship:

$$\boldsymbol{R} = \exp([\boldsymbol{\phi}]_\times) \tag{1a}$$

$$\boldsymbol{\phi} = \mathrm{Log}(\boldsymbol{R}) \tag{1b}$$

where $[\ ]_\times$ represents skew-symmetric mapping, which maps $\boldsymbol{\phi} = [\phi_x, \phi_y, \phi_z]^\top \in \mathbb{R}^3$ to an skew-symmetric matrix

$$[\boldsymbol{\phi}]_\times = \begin{bmatrix} 0 & -\phi_z & \phi_y \\ \phi_z & 0 & -\phi_x \\ -\phi_y & \phi_x & 0 \end{bmatrix}. \tag{2}$$

When the rotation angle, $\|\boldsymbol{\phi}\| \ll 1^\circ$, the approximate relationship exists as follows [2]:

$$\boldsymbol{R} \approx \mathbf{1}_3 + [\boldsymbol{\phi}]_\times \tag{3}$$

where, $\mathbf{1}_3$ is an 3-D identity matrix. Euler angles in this article are defined as the intrinsic z-y-x rotation order, i.e., $\theta_x, \theta_y, \theta_z$ represent yaw, pith, and roll angle, respectively, then having the following relationship between $\boldsymbol{R}$ and $\boldsymbol{\theta}$:

$$\boldsymbol{R} = \begin{bmatrix} c_x c_y & c_x s_y s_z - c_z s_x & s_x s_z + c_x c_z s_y \\ c_y s_x & c_x c_z + s_x s_y s_z & c_z s_x s_y - c_x s_z \\ -s_y & c_y s_z & c_y c_z \end{bmatrix} \tag{4}$$

here, c_i and s_i represent $\cos(\theta_i)$ and $\sin(\theta_i)$, respectively $(i = x, y, z)$.

IV. Subsystem

This section will introduce the workflow of the three subsystems: EAS, VIS, and LS, respectively. For the convenience of description, it is assumed that all sensor coordinate frames coincide with $\mathcal{L}$, which can be achieved by simply offline calibrating external parameters.

A. Encoder-AHRS Subsystem

1) Description: Four encoders are, respectively, installed on the shaft of each wheel to measure the rotational speed of the wheel in real-time. The *forward kinematics* module converts the speeds into the linear velocity v^{kin} and angular velocity ω^{kin} of the robot. AHRS, composed of an accelerometer, gyroscope, magnetometer and on-board micro processor, is an orientation estimator providing attitude information of the robot, including yaw, pitch, and roll angle. The *velocity decomposition* module decomposes a velocity expressed in $\mathcal{L}$ into a velocity expressed in $\mathcal{G}$ by the following formula:

$$\boldsymbol{v} = \boldsymbol{R} \begin{bmatrix} v^{\text{kin}} \\ 0 \\ 0 \end{bmatrix}. \tag{5}$$

Then, the *velocity integration* module integrates $\boldsymbol{v}$ to obtain the position of the robot, $\boldsymbol{p} \in \mathbb{R}^3$

$$\boldsymbol{p} = \int_0^t \boldsymbol{v} \mathrm{d}t. \tag{6}$$

Due to measurement and other uncertainties, what is calculated above is actually estimates of position, attitude and global velocity at the time t_k, denoted as $\{\tilde{\boldsymbol{p}}^{\text{eas}}_k, \tilde{\boldsymbol{R}}^{\text{eas}}_k, \tilde{\boldsymbol{v}}^{\text{eas}}_k\}$.

2) Output Model: In fact, rugged or slippery ground is likely to destroy certain assumptions of the robot's kinematics [35], and magnetometer signals may be corrupted by some magnetic disturbances (e.g., ferromagnetic and permanent magnetic materials) [36]. These uncontrollable situations will inevitably lead to the accumulation of pose errors with time and driving distance. Therefore, in order to reasonably describe the uncertainty of the system output, we choose to model the position increment (PI) and the attitude increment (AI). Let the measurements of PI and AI during time interval $[t_{k-1}, t_k]$ (about 50 ms) be $\tilde{\boldsymbol{\Delta}}^{\text{eas}}_{p,k} \triangleq \tilde{\boldsymbol{R}}^{\text{eas}\top}_{k-1}(\tilde{\boldsymbol{p}}^{\text{eas}}_k - \tilde{\boldsymbol{p}}^{\text{eas}}_{k-1}) \in \mathbb{R}^3$ and $\tilde{\boldsymbol{\Delta}}^{\text{eas}}_{R,k} \triangleq \tilde{\boldsymbol{R}}^{\text{eas}\top}_{k-1} \tilde{\boldsymbol{R}}^{\text{eas}}_k \in \text{SO}(3)$, respectively, and their measurement process are modeled as follows:

$$\tilde{\boldsymbol{\Delta}}^{\text{eas}}_{p,k} = \boldsymbol{R}^\top_{k-1}(\boldsymbol{p}_k - \boldsymbol{p}_{k-1}) + \boldsymbol{w}^{\text{eas}}_{\Delta_p,k} \tag{7a}$$

$$\tilde{\boldsymbol{\Delta}}^{\text{eas}}_{R,k} = \exp\left(\left[\boldsymbol{w}^{\text{eas}}_{\Delta_R,k}\right]_\times\right)\boldsymbol{R}^\top_{k-1}\boldsymbol{R}_k \tag{7b}$$

where measurement noises are assumed to be Gaussian white noise, i.e., $\boldsymbol{w}^{\text{eas}}_{\Delta_p,k} \sim \mathcal{N}(\mathbf{0}_3, \boldsymbol{\Sigma}^{\text{eas}}_{\Delta_p,k})$, $\boldsymbol{w}^{\text{eas}}_{\Delta_R,k} \sim \mathcal{N}(\mathbf{0}_3, \boldsymbol{\Sigma}^{\text{eas}}_{\Delta_R,k})$, in which the covariance can be determined by experiments.

In addition, since the estimate of roll and pitch from AHRS is obtained with reference to the gravity vector, it can be seen from (4) and (5) that the third component of $\tilde{\boldsymbol{v}}^{\text{eas}}$, namely GVV $\tilde{v}^{\text{eas}}_z$, is drift-free and more in line with the Gaussian white noise assumption than the first and second components of $\tilde{\boldsymbol{v}}^{\text{eas}}$. The above analysis shows that in the fusion center, $\tilde{v}^{\text{eas}}_z$ should be preferentially selected to participate in the fusion. We model the GVV measurement process as follows:

$$\tilde{v}^{\text{eas}}_z = v_z + w^{\text{eas}}_{v_z} \tag{8}$$

where, $w_{v_z} \sim \mathcal{N}(0, \sigma^{\text{eas}2}_{v_z})$. To further model the uncertainty that may be introduced by the tire slippage and rough ground, the covariance $\sigma^{\text{eas}2}_{v_z}$ will be adaptively adjusted according to the following adjustment strategy:

$$\sigma^{\text{eas}}_{v_z} = \begin{cases} s_1\sigma_0, & \text{if } |\tilde{\omega}^{\text{kin}} - \tilde{\omega}^{\text{ahrs}}| < \tau_1 \\ \sigma_0, & \text{if } \tau_1 \le |\tilde{\omega}^{\text{kin}} - \tilde{\omega}^{\text{ahrs}}| < \tau_2 \\ s_2|\tilde{\omega}^{\text{kin}} - \tilde{\omega}^{\text{ahrs}}|\sigma_0, & \text{if } |\tilde{\omega}^{\text{kin}} - \tilde{\omega}^{\text{ahrs}}| \ge \tau_2 \end{cases} \tag{9}$$

where, $\tilde{\omega}^{\text{kin}}$, $\tilde{\omega}^{\text{ahrs}}$ are the angular velocity measurements from kinematics and AHRS, respectively. τ_1, τ_2 are two different thresholds (set to 0.03, 0.3 for us), and s_1, s_2 are two different scale coefficients (set to 10^{-2}, 10^2 for us). σ_0 is the base value, whose value is set to 0.05 in our experiments.

B. Visual-Inertial Subsystem

1) Description: We adapt the processing pipeline from [33] for VIS. The *feature tracking* module detects the visual features for each camera frame using corner detector [37], and tracks features in the previous frame using Kanade–Lucas–Tomasi algorithm [38], and matches features among the left image and right image. The *IMU pre-integration* module integrates inertial data between the previous frame and current frame to obtain pre-integration measurement (i.e., relative position, velocity, and rotation) and their uncertainties according to the algorithm in [39]. The *sliding-window optimization* module is responsible for constructing a factor graph using the pinhole camera model and pre-integration measurement model, and solving state variables using Ceres solver,[1] and the size of the factor graph is limited to a sliding window (e.g., the size is set to 10). There are four types of factors in the factor graph: re-projection factor between two frames, the re-projection factor between the left and right camera, the

[1] http://ceres-solver.org/

pre-integration factor and the marginalization factor, and their detailed definition and construction can be found at [16], [33]. The state variables estimated in the factor graph are

$$X = \{\boldsymbol{p}_i, \boldsymbol{R}_i, \boldsymbol{v}_i, \boldsymbol{b}_{f,i}, \boldsymbol{b}_{\omega,i}, d_j\} \qquad i = 0, \ldots, M; \quad j = 0, \ldots, N \tag{10}$$

where M and N are the number of image frames in the sliding window and the number of observed features, respectively. $\boldsymbol{p}_i, \boldsymbol{R}_i, \boldsymbol{v}_i, \boldsymbol{b}_{f,i}, \boldsymbol{b}_{\omega,i}$, respectively, represent the position, attitude, velocity, accelerometer bias and gyroscope bias at the time of the ith frame. d_j represents the depth of the jth feature observed in the first frame. After solving the sliding-window factor graph, the outputs of VIS at time t_k are the estimates of position, attitude, velocity, accel. bias, and gyro. bias, denoted as $\{\tilde{\boldsymbol{p}}_k^{\text{vis}}, \tilde{\boldsymbol{R}}_k^{\text{vis}}, \tilde{\boldsymbol{v}}_k^{\text{vis}}, \tilde{\boldsymbol{b}}_{f,k}^{\text{vis}}, \tilde{\boldsymbol{b}}_{\omega,k}^{\text{vis}}\}$.

It should be noted here that, $\tilde{\boldsymbol{v}}^{\text{vis}}$ is the estimate of global velocity, which drifts gradually with driving time and distance. The estimate of the velocity expressed in $\mathcal{L}$, denoted as $\tilde{\boldsymbol{v}}^{\text{vis},\mathcal{L}}$, can be obtain using the following formula:

$$\tilde{\boldsymbol{v}}^{\text{vis},\mathcal{L}} = \tilde{\boldsymbol{R}}^{\text{vis}\top} \tilde{\boldsymbol{v}}^{\text{vis}}. \tag{11}$$

2) Output Model: Like EAS, we build the measurement model of PI and AI during time interval $[t_{k-1}, t_k]$ (about 100 ms) as follows:

$$\tilde{\boldsymbol{\Delta}}_{p,k}^{\text{vis}} = \boldsymbol{R}_{k-1}^{\top}(\boldsymbol{p}_k - \boldsymbol{p}_{k-1}) + \boldsymbol{w}_{\Delta_p,k}^{\text{vis}} \tag{12a}$$

$$\tilde{\boldsymbol{\Delta}}_{R,k}^{\text{vis}} = \exp\left(\left[\boldsymbol{w}_{\Delta_R,k}^{\text{vis}}\right]_{\times}\right)\boldsymbol{R}_{k-1}^{\top}\boldsymbol{R}_k \tag{12b}$$

where $\boldsymbol{w}_{\Delta_p,k}^{\text{vis}} \sim \mathcal{N}(\boldsymbol{0}_3, \boldsymbol{\Sigma}_{\Delta_p,k}^{\text{vis}})$, $\boldsymbol{w}_{\Delta_R,k}^{\text{vis}} \sim \mathcal{N}(\boldsymbol{0}_3, \boldsymbol{\Sigma}_{\Delta_R,k}^{\text{vis}})$ in which the covariance can also be determined by experiments. For the LV, the measurement model of LV is built as follows:

$$\tilde{\boldsymbol{v}}^{\text{vis},\mathcal{L}} = \boldsymbol{R}^{\top}\boldsymbol{v} + \boldsymbol{w}_{v}^{\text{vis},\mathcal{L}} \tag{13}$$

where, $\boldsymbol{w}_{v}^{\text{vis},\mathcal{L}} \sim \mathcal{N}(\boldsymbol{0}_3, \boldsymbol{\Sigma}_{v}^{\text{vis},\mathcal{L}})$.

C. LiDAR Subsystem

1) Description: The LS is adapted from the processing pipeline of [31]. In the *feature extraction* module, motion compensation algorithm [21] based on IMU data is applied to a scan of raw points to remove the distortion caused by LiDAR own motion, next, the points are filtered and divided into ground points and the other large object points using the ground plane estimation method [40] and the fast range segmentation method [41], lastly, edge and plane feature points are extracted according to the roughness of each point. In the *scan to scan* module, point-to-edge-line and point-to-plane-patch matching are performed to estimate the LiDAR pose increment between two consecutive scans composed of edge and plane points. The process operates at a frequency of about 20 Hz, and the detailed procedures of scan-to-scan matching can be found in [42]. The *scan to submap* module matches the current scan to submap (i.e., the previous scans transformed into $\mathcal{G}$ whose sensor poses are within 150 m of the current position of the sensor) to further refine the pose estimates, and this procedure runs at a lower frequency of about 2 Hz. The scan-to-submap problem can be formulated as the following minimization:

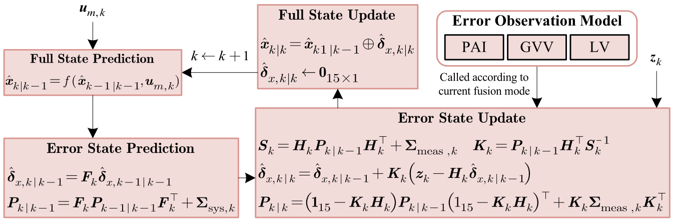
$\boldsymbol{u}_{m,k}$
Full State Prediction
$\hat{\boldsymbol{x}}_{k|k-1} = f(\hat{\boldsymbol{x}}_{k-1|k-1}, \boldsymbol{u}_{m,k})$
$k \leftarrow k+1$
Full State Update
$\hat{\boldsymbol{x}}_{k|k} = \hat{\boldsymbol{x}}_{k1|k-1} \oplus \hat{\boldsymbol{\delta}}_{x,k|k}$
$\hat{\boldsymbol{\delta}}_{x,k|k} \leftarrow \boldsymbol{0}_{15\times1}$
Error Observation Model
PAI
GVV
LV
Called according to current fusion mode
$\boldsymbol{z}_k$
Error State Prediction
$\hat{\boldsymbol{\delta}}_{x,k|k-1} = \boldsymbol{F}_k\hat{\boldsymbol{\delta}}_{x,k-1|k-1}$
$\boldsymbol{P}_{k|k-1} = \boldsymbol{F}_k\boldsymbol{P}_{k-1|k-1}\boldsymbol{F}_k^{\top} + \boldsymbol{\Sigma}_{\text{sys},k}$
Error State Update
$\boldsymbol{S}_k = \boldsymbol{H}_k\boldsymbol{P}_{k|k-1}\boldsymbol{H}_k^{\top} + \boldsymbol{\Sigma}_{\text{meas},k} \quad \boldsymbol{K}_k = \boldsymbol{P}_{k|k-1}\boldsymbol{H}_k^{\top}\boldsymbol{S}_k^{-1}$
$\hat{\boldsymbol{\delta}}_{x,k|k} = \hat{\boldsymbol{\delta}}_{x,k|k-1} + \boldsymbol{K}_k(\boldsymbol{z}_k - \boldsymbol{H}_k\hat{\boldsymbol{\delta}}_{x,k|k-1})$
$\boldsymbol{P}_{k|k} = (\boldsymbol{1}_{15} - \boldsymbol{K}_k\boldsymbol{H}_k)\boldsymbol{P}_{k|k-1}(\boldsymbol{1}_{15} - \boldsymbol{K}_k\boldsymbol{H}_k)^{\top} + \boldsymbol{K}_k\boldsymbol{\Sigma}_{\text{meas},k}\boldsymbol{K}_k^{\top}$

Fig. 3. Flowchart of variable fusion mode state estimator.

$$\tilde{\boldsymbol{p}}^{\text{ls}}, \tilde{\boldsymbol{R}}^{\text{ls}} = \underset{\boldsymbol{p},\boldsymbol{R}}{\arg\min}\ \frac{1}{2}\left\{\sum_{i=1}^{N^e} r_i^{e\,2} + \sum_{j=1}^{N^p} r_j^{p\,2}\right\} \tag{14}$$

where N^e and N^p are the number of the edge and plane points in the current scan, respectively. r_i^e and r_j^p are the point-to-edge-line and point-to-plane-patch distance error, respectively, defined as follows:

$$r_i^e = w_i^e \frac{\left\|\left(\boldsymbol{R}\tilde{\boldsymbol{p}}_i^e + \boldsymbol{p} - \tilde{\boldsymbol{p}}_u^e\right) \times \left(\boldsymbol{R}\tilde{\boldsymbol{p}}_i^e + \boldsymbol{p} - \tilde{\boldsymbol{p}}_v^e\right)\right\|}{\left\|\tilde{\boldsymbol{p}}_u^e - \tilde{\boldsymbol{p}}_v^e\right\|}$$

$$r_j^p = w_j^p \left\|\left(\boldsymbol{R}\tilde{\boldsymbol{p}}_j^p + \boldsymbol{p} - \tilde{\boldsymbol{p}}_w^p\right)^{\top} \frac{\left(\tilde{\boldsymbol{p}}_u^p - \tilde{\boldsymbol{p}}_w^p\right) \times \left(\tilde{\boldsymbol{p}}_v^p - \tilde{\boldsymbol{p}}_w^p\right)}{\left\|\left(\tilde{\boldsymbol{p}}_u^p - \tilde{\boldsymbol{p}}_w^p\right) \times \left(\tilde{\boldsymbol{p}}_v^p - \tilde{\boldsymbol{p}}_w^p\right)\right\|}\right\|$$

here, w_i^e and w_j^p are the weights. $\tilde{\boldsymbol{p}}_i^e$ and $\tilde{\boldsymbol{p}}_j^p$ are ith edge point and jth plane point in the current scan, respectively. $\tilde{\boldsymbol{p}}_u^e$ and $\tilde{\boldsymbol{p}}_v^e$ are the edge points in the submap, which define the edge line associated with $\tilde{\boldsymbol{p}}_i^e$. $\tilde{\boldsymbol{p}}_u^p$, $\tilde{\boldsymbol{p}}_v^p$ and $\tilde{\boldsymbol{p}}_w^p$ are the plane points in the submap, which define the plane patch associated with $\tilde{\boldsymbol{p}}_j^p$. Finally, the output of LS at time t_k are the estimates of position, attitude, denoted as $\{\tilde{\boldsymbol{p}}_k^{\text{ls}}, \tilde{\boldsymbol{R}}_k^{\text{ls}}\}$.

2) Output Model: Like EAS and VIS, we build the measurement model of PI and AI during time interval $[t_{k-1}, t_k]$ (about 500 ms) as follows:

$$\tilde{\boldsymbol{\Delta}}_{p,k}^{\text{ls}} = \boldsymbol{R}_{k-1}^{\top}(\boldsymbol{p}_k - \boldsymbol{p}_{k-1}) + \boldsymbol{w}_{\Delta_p,k}^{\text{ls}} \tag{15a}$$

$$\tilde{\boldsymbol{\Delta}}_{R,k}^{\text{ls}} = \exp\left(\left[\boldsymbol{w}_{\Delta_R,k}^{\text{ls}}\right]_{\times}\right)\boldsymbol{R}_{k-1}^{\top}\boldsymbol{R}_k \tag{15b}$$

where $\boldsymbol{w}_{\Delta_p,k}^{\text{ls}} \sim \mathcal{N}(\boldsymbol{0}_3, \boldsymbol{\Sigma}_{\Delta_p,k}^{\text{ls}})$, $\boldsymbol{w}_{\Delta_R,k}^{\text{ls}} \sim \mathcal{N}(\boldsymbol{0}_3, \boldsymbol{\Sigma}_{\Delta_R,k}^{\text{ls}})$ in which the covariance can also be determined by experiments.

V. Variable Fusion Mode Estimator

Fig. 3 presents our designed variable fusion mode estimator based on ES-EKF. It consists of three phases: prediction, observation, and updating.

A. State Variables Definition

Table I lists all state variables and their symbols in the proposed state estimator. Full state $\boldsymbol{x}$ and error state $\boldsymbol{\delta}_x$ are defined as follows:

$$\boldsymbol{x} \triangleq \{\boldsymbol{p}, \boldsymbol{R}, \boldsymbol{v}, \boldsymbol{b}_f, \boldsymbol{b}_\omega\} \tag{16a}$$

$$\boldsymbol{\delta}_x \triangleq \left[\boldsymbol{\delta}_p^{\top}\ \boldsymbol{\delta}_a^{\top}\ \boldsymbol{\delta}_v^{\top}\ \boldsymbol{\delta}_{bf}^{\top}\ \boldsymbol{\delta}_{b\omega}^{\top}\right]^{\top} \tag{16b}$$

where the error of each variable is defined as

$$\boldsymbol{\delta}_p \triangleq \boldsymbol{p} - \hat{\boldsymbol{p}} \tag{17a}$$

TABLE I

ALL STATE VARIABLES IN STATE ESTIMATOR WITH VARIABLE MODE

Magnitude	True	Estimate	Space	Frame
Full state	$\boldsymbol{x}$	$\hat{\boldsymbol{x}}$		
Position	$\boldsymbol{p}$	$\hat{\boldsymbol{p}}$	$\mathbb{R}^3$	G
Attitude (Rotation matrix)	$\boldsymbol{R}$	$\hat{\boldsymbol{R}}$	$SO(3)$	G
Velocity	$\boldsymbol{v}$	$\hat{\boldsymbol{v}}$	$\mathbb{R}^3$	G
Accelerometer bias	$\boldsymbol{b}_f$	$\hat{\boldsymbol{b}}_f$	$\mathbb{R}^3$	L
Gyroscope bias	$\boldsymbol{b}_\omega$	$\hat{\boldsymbol{b}}_\omega$	$\mathbb{R}^3$	L
Error state	$\boldsymbol{\delta}_x$	$\hat{\boldsymbol{\delta}}_x$	$\mathbb{R}^{15}$	
Position error	$\boldsymbol{\delta}_p$	$\hat{\boldsymbol{\delta}}_p$	$\mathbb{R}^3$	G
Attitude error	$\boldsymbol{\delta}_a$	$\hat{\boldsymbol{\delta}}_a$	$\mathbb{R}^3$	G
Velocity error	$\boldsymbol{\delta}_v$	$\hat{\boldsymbol{\delta}}_v$	$\mathbb{R}^3$	G
Accelerometer bias error	$\boldsymbol{\delta}_{bf}$	$\hat{\boldsymbol{\delta}}_{bf}$	$\mathbb{R}^3$	L
Gyroscope bias error	$\boldsymbol{\delta}_{b\omega}$	$\hat{\boldsymbol{\delta}}_{b\omega}$	$\mathbb{R}^3$	L

$$\boldsymbol{\delta}_a \triangleq \mathrm{Log}(\boldsymbol{R}\hat{\boldsymbol{R}}^\top) \tag{17b}$$

$$\boldsymbol{\delta}_v \triangleq \boldsymbol{v} - \hat{\boldsymbol{v}} \tag{17c}$$

$$\boldsymbol{\delta}_{bf} \triangleq \boldsymbol{b}_f - \hat{\boldsymbol{b}}_f \tag{17d}$$

$$\boldsymbol{\delta}_{b\omega} \triangleq \boldsymbol{b}_\omega - \hat{\boldsymbol{b}}_\omega. \tag{17e}$$

B. IMU-Driven State Kinematics

1) True-State Kinematics: Based on the general MEMS-IMU measurement model [16], [19], in which the measurements of specific force and angular velocity, denoted as $\boldsymbol{f}_m$ and $\boldsymbol{\omega}_m$, are affected by accelerometer bias $\boldsymbol{b}_f$, gyroscope bias $\boldsymbol{b}_\omega$ and addictive noise, the true-state kinematics are modeled as

$$\begin{cases} \dot{\boldsymbol{p}} = \boldsymbol{v} \\ \dot{\boldsymbol{R}} = \boldsymbol{R}[\boldsymbol{\omega}_m - \boldsymbol{b}_\omega - \boldsymbol{n}_\omega]_\times \\ \dot{\boldsymbol{v}} = \boldsymbol{R}\big(\boldsymbol{f}_m - \boldsymbol{b}_f - \boldsymbol{n}_f\big) + \mathbf{g} \\ \dot{\boldsymbol{b}}_f = \boldsymbol{\tau}_{bf} \\ \dot{\boldsymbol{b}}_\omega = \boldsymbol{\tau}_{b\omega} \end{cases} \tag{18}$$

where, additive noise $\boldsymbol{n}_f, \boldsymbol{n}_\omega$ in accelerometer and gyroscope measurements are assumed to be Gaussian white noise, i.e., $\boldsymbol{n}_f \sim \mathcal{N}(\mathbf{0}_{3\times1}, \sigma_f^2\mathbf{1}_3), \boldsymbol{n}_\omega \sim \mathcal{N}(\mathbf{0}_{3\times1}, \sigma_\omega^2\mathbf{1}_3)$. $\boldsymbol{b}_f, \boldsymbol{b}_\omega$ are modeled as a random walk, whose derivatives are Gaussian white noise, i.e., $\boldsymbol{\tau}_{bf} \sim \mathcal{N}(\mathbf{0}_{3\times1}, \sigma_{bf}^2\mathbf{1}_3)$, $\boldsymbol{\tau}_{b\omega} \sim \mathcal{N}(\mathbf{0}_{3\times1}, \sigma_{b\omega}^2\mathbf{1}_3)$. $\mathbf{g}$ is the gravitational acceleration constant in $\mathcal{G}$.

2) Estimate-State Kinematics: Neglecting Gaussian white noise in (18), the estimate-state kinematics are obtained as follows:

$$\begin{cases} \dot{\hat{\boldsymbol{p}}} = \hat{\boldsymbol{v}} \\ \dot{\hat{\boldsymbol{R}}} = \hat{\boldsymbol{R}}\big[\boldsymbol{\omega}_m - \hat{\boldsymbol{b}}_\omega\big]_\times \\ \dot{\hat{\boldsymbol{v}}} = \hat{\boldsymbol{R}}\big(\boldsymbol{f}_m - \hat{\boldsymbol{b}}_f\big) + \mathbf{g} \\ \dot{\hat{\boldsymbol{b}}}_f = \mathbf{0}_{3\times1} \\ \dot{\hat{\boldsymbol{b}}}_\omega = \mathbf{0}_{3\times1}. \end{cases} \tag{19}$$

3) First-Order Error-State Kinematics: According to the error state definition (17), comparing true-state kinematics (18) and estimate-state kinematics (19), the first-order kinematics of the error state are obtained

$$\dot{\boldsymbol{\delta}}_x = \boldsymbol{A}\boldsymbol{\delta}_x + \boldsymbol{B}\boldsymbol{n}_{\mathrm{sys}} \tag{20}$$

where

$$\boldsymbol{A} = \begin{bmatrix} \mathbf{0}_3 & \mathbf{1}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & -\hat{\boldsymbol{R}} \\ \mathbf{0}_3 & -\big[\hat{\boldsymbol{R}}\big(\boldsymbol{f}_m - \boldsymbol{b}_f\big)\big]_\times & \mathbf{0}_3 & -\hat{\boldsymbol{R}} & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 \end{bmatrix}$$

$$\boldsymbol{B} = \begin{bmatrix} \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & -\hat{\boldsymbol{R}} & \mathbf{0}_3 & \mathbf{0}_3 \\ -\hat{\boldsymbol{R}} & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{1}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{1}_3 \end{bmatrix}, \quad \boldsymbol{n}_{\mathrm{sys}} = \begin{bmatrix} \boldsymbol{n}_f \\ \boldsymbol{n}_\omega \\ \boldsymbol{\tau}_{bf} \\ \boldsymbol{\tau}_{b\omega} \end{bmatrix}.$$

C. Prediction Using IMU Data

1) Full State Prediction: Suppose the posterior estimate of $\boldsymbol{x}$ at time t_{k-1} is $\hat{\boldsymbol{x}}_{k-1|k-1}$, then the prediction of $\boldsymbol{x}$ at time $t_k (t_k = t_{k-1} + \Delta_t)$, $\hat{\boldsymbol{x}}_{k|k-1}$, can be obtained by integrating the kinematics (19). The prediction equation is abbreviated as

$$\hat{\boldsymbol{x}}_{k|k-1} = f(\hat{\boldsymbol{x}}_{k-1|k-1}, \boldsymbol{u}_{m,k}) \tag{21}$$

where $\boldsymbol{u}_{m,k} \triangleq [\boldsymbol{f}_{m,k-1}^\top \ \boldsymbol{\omega}_{m,k-1}^\top]^\top$, and the detailed formula is

$$\begin{cases} \hat{\boldsymbol{p}}_{k|k-1} = \hat{\boldsymbol{p}}_{k-1|k-1} + \hat{\boldsymbol{v}}_{k-1|k-1}\Delta_t \\ \qquad\qquad + \dfrac{1}{2}\Big(\hat{\boldsymbol{R}}_{k-1|k-1}\big(\boldsymbol{f}_{m,k-1} - \hat{\boldsymbol{b}}_{f,k-1|k-1}\big) + \mathbf{g}\Big)\Delta_t^2 \\ \hat{\boldsymbol{R}}_{k|k-1} = \hat{\boldsymbol{R}}_{k-1|k-1}\exp\Big(\big[\boldsymbol{\omega}_{m,k-1} - \hat{\boldsymbol{b}}_{\omega,k-1}\big]_\times\Delta_t\Big) \\ \hat{\boldsymbol{v}}_{k|k-1} = \hat{\boldsymbol{v}}_{k-1|k-1} \\ \qquad\qquad + \Big(\hat{\boldsymbol{R}}_{k-1|k-1}\big(\boldsymbol{f}_{m,k-1} - \hat{\boldsymbol{b}}_{f,k-1|k-1}\big) + \mathbf{g}\Big)\Delta_t \\ \hat{\boldsymbol{b}}_{f,k|k-1} = \hat{\boldsymbol{b}}_{f,k-1|k-1} \\ \hat{\boldsymbol{b}}_{\omega,k|k-1} = \hat{\boldsymbol{b}}_{\omega,k-1|k-1}. \end{cases}$$

2) Error State Prediction: According to the error-state kinematics (20), a prediction model of $\boldsymbol{\delta}_x$ is established below

$$\boldsymbol{\delta}_{x,k} = \boldsymbol{F}_k\boldsymbol{\delta}_{x,k-1} + \boldsymbol{w}_{\mathrm{sys},k} \tag{22}$$

where $\boldsymbol{F}_k \in \mathbb{R}^{15\times15}$ is error transition matrix; $\boldsymbol{w}_{\mathrm{sys},k} \in \mathbb{R}^{15}$ is the process noise subject to the assumption of Gaussian white noise. i.e.,

$$\begin{aligned} \boldsymbol{F}_k &= e^{\boldsymbol{A}_{k-1}\Delta_t} \approx \mathbf{1}_{15} + \boldsymbol{A}_{k-1}\Delta_t + \frac{1}{2}(\boldsymbol{A}_{k-1}\Delta_t)^2 \\ \boldsymbol{w}_{\mathrm{sys},k} &\sim \mathcal{N}\big(\mathbf{0}_{15\times1}, \boldsymbol{\Sigma}_{\mathrm{sys},k}\big) \\ \boldsymbol{\Sigma}_{\mathrm{sys},k} &= \boldsymbol{B}_{k-1}\mathrm{diag}\big(\sigma_f^2\mathbf{1}_3, \sigma_\omega^2\mathbf{1}_3, \sigma_{bf}^2\mathbf{1}_3, \sigma_{b\omega}^2\mathbf{1}_3\big)\boldsymbol{B}_{k-1}^\top. \end{aligned}$$

Let the error estimate and its covariance at time t_{k-1} to be $\hat{\boldsymbol{\delta}}_{x,k-1|k-1}$ and $\boldsymbol{P}_{k-1|k-1}$, respectively. Then, according to the prediction formula of Kalman filtering, the predicted error and its covariance at time t_k, respectively, denoted as $\hat{\boldsymbol{\delta}}_{x,k|k-1}$ and $\boldsymbol{P}_{k|k-1}$, can be propagated as follows:

$$\hat{\boldsymbol{\delta}}_{x,k|k-1} = \boldsymbol{F}_k\hat{\boldsymbol{\delta}}_{x,k-1|k-1} \tag{23a}$$

$$\boldsymbol{P}_{k|k-1} = \boldsymbol{F}_k\boldsymbol{P}_{k-1|k-1}\boldsymbol{F}_k^\top + \boldsymbol{\Sigma}_{\mathrm{sys},k}. \tag{23b}$$

D. Error Observation Models for Subsystems

1) Position and Attitude Increment: For the convenience of description, the measurement models of PI and AI in (7), (12), and (24) are uniformly denoted as

$$\begin{aligned}\tilde{\boldsymbol{\Delta}}_{p,k} &= \boldsymbol{R}_{k-1}^{\top}(\boldsymbol{p}_k - \boldsymbol{p}_{k-1}) + \boldsymbol{w}_{\Delta_p,k} && \text{(24a)}\\ \tilde{\boldsymbol{\Delta}}_{R,k} &= \exp([\boldsymbol{w}_{\Delta_R,k}]_{\times})\boldsymbol{R}_{k-1}^{\top}\boldsymbol{R}_k && \text{(24b)}\end{aligned}$$

where $\boldsymbol{w}_{\Delta_p,k} \sim \mathcal{N}(\boldsymbol{0}_3, \boldsymbol{\Sigma}_{\Delta_p,k})$, $\boldsymbol{w}_{\Delta_R,k} \sim \mathcal{N}(\boldsymbol{0}_3, \boldsymbol{\Sigma}_{\Delta_R,k})$. Consider the error definition (17) and the approximate relationship (3), we have

$$\begin{aligned}&\boldsymbol{R}_{k-1}^{\top}(\boldsymbol{p}_k - \boldsymbol{p}_{k-1})\\ &\quad\approx \hat{\boldsymbol{R}}_{k-1}^{\top}(\hat{\boldsymbol{p}}_k - \hat{\boldsymbol{p}}_{k-1}) + \hat{\boldsymbol{R}}_{k-1}^{\top}\boldsymbol{\delta}_{\boldsymbol{p},k} - \hat{\boldsymbol{R}}_{k-1}^{\top}\boldsymbol{\delta}_{\boldsymbol{p},k-1}\\ &\qquad + \hat{\boldsymbol{R}}_{k-1}^{\top}[\hat{\boldsymbol{p}}_k - \hat{\boldsymbol{p}}_{k-1}]_{\times}\boldsymbol{\delta}_{a,k-1} && \text{(25a)}\\ &\exp([\boldsymbol{w}_{\Delta_R,k}]_{\times})\boldsymbol{R}_{k-1}^{\top}\boldsymbol{R}_k\\ &\quad\approx \exp\Big(\Big[\hat{\boldsymbol{R}}_{k-1}^{\top}\big(\boldsymbol{\delta}_{a,k} - \boldsymbol{\delta}_{a,k-1}\big) + \boldsymbol{w}_{\Delta_R,k}\Big]_{\times}\Big)\hat{\boldsymbol{R}}_{k-1}^{\top}\hat{\boldsymbol{R}}_k. && \text{(25b)}\end{aligned}$$

Then, the error observation model of PAI can be established as

$$z_{\mathrm{PAI},k} = \boldsymbol{H}_{\mathrm{PAI},k}\boldsymbol{\delta}_{x,k} + \boldsymbol{w}_{\mathrm{PAI},k} \qquad (26)$$

where

$$\begin{aligned}z_{\mathrm{PAI},k} &\triangleq \begin{bmatrix}\tilde{\boldsymbol{\Delta}}_{R,k} - \hat{\boldsymbol{R}}_{k-1}^{\top}\big(\hat{\boldsymbol{p}}_k - \hat{\boldsymbol{p}}_{k-1}\big)\\ \mathrm{Log}\Big(\tilde{\boldsymbol{\Delta}}_{R,k}\hat{\boldsymbol{R}}_k^{\top}\hat{\boldsymbol{R}}_{k-1}\Big)\end{bmatrix}\\ \boldsymbol{H}_{\mathrm{PAI}} &= \begin{bmatrix}\hat{\boldsymbol{R}}_{k-1}^{\top} & \boldsymbol{0}_3 & \boldsymbol{0}_3 & \boldsymbol{0}_3 & \boldsymbol{0}_3\\ \boldsymbol{0}_3 & \hat{\boldsymbol{R}}_{k-1}^{\top} & \boldsymbol{0}_3 & \boldsymbol{0}_3 & \boldsymbol{0}_3\end{bmatrix}\\ \boldsymbol{w}_{\mathrm{PAI},k} &\sim \mathcal{N}\big(\boldsymbol{0}_{6\times 1},\ \mathrm{diag}(\boldsymbol{\Sigma}_{11,k}, \boldsymbol{\Sigma}_{22,k})\big)\\ \boldsymbol{\Sigma}_{11,k} &= \boldsymbol{\Sigma}_{\Delta_p,k} + \hat{\boldsymbol{R}}_{k-}^{\top}\boldsymbol{P}_{p,k-1}\hat{\boldsymbol{R}}_{k-1}\\ &\quad \times \hat{\boldsymbol{R}}_{k-1}^{\top}[\hat{\boldsymbol{p}}_k - \hat{\boldsymbol{p}}_{k-1}]_{\times}\boldsymbol{P}_{a,k-1}[\hat{\boldsymbol{p}}_k - \hat{\boldsymbol{p}}_{k-1}]_{\times}\hat{\boldsymbol{R}}_{k-1}\\ \boldsymbol{\Sigma}_{22,k} &= \boldsymbol{\Sigma}_{\Delta_p,k} + \hat{\boldsymbol{R}}_{k-1}^{\top}\boldsymbol{P}_{a,k-1}\hat{\boldsymbol{R}}_{k-1}\end{aligned}$$

here, $\boldsymbol{P}_{p,k-1}$ and $\boldsymbol{P}_{a,k-1}$ are, respectively, position error covariance and attitude error covariance at time t_{k-1}, extracted from total error covariance $\boldsymbol{P}_{k-1}$.

2) Global Vertical Velocity: Combining (8) and (17c), the following equation is obtained

$$\tilde{v}_z^{\mathrm{eas}} = v_z + w_{v_z}^{\mathrm{eas}} = \delta_{v_z} + \hat{v}_z + w_{v_z}^{\mathrm{eas}}. \qquad (27)$$

Let $z_{\mathrm{GVV},k} \triangleq \tilde{v}_{z,k}^{\mathrm{eas}} - \hat{v}_{z,k}$, the error observation model of GVV is established as

$$z_{\mathrm{GVV},k} = \boldsymbol{H}_{\mathrm{GVV},k}\boldsymbol{\delta}_{x,k} + w_{v_z,k}^{\mathrm{eas}} \qquad (28)$$

where

$$\boldsymbol{H}_{\mathrm{GVV},k} = \begin{bmatrix}\boldsymbol{0}_{1\times 8} & 1 & \boldsymbol{0}_{1\times 6}\end{bmatrix}.$$

3) Local Velocity: Combining (13), (17), and (3), the following approximate relationship is obtained:

$$\tilde{\boldsymbol{v}}^{\mathrm{vis},\mathcal{L}} - \hat{\boldsymbol{R}}^{\top}\hat{\boldsymbol{v}} \approx \hat{\boldsymbol{R}}^{\top}[\hat{\boldsymbol{v}}]_{\times}\boldsymbol{\delta}_a + \hat{\boldsymbol{R}}^{\top}\boldsymbol{\delta}_v + \boldsymbol{w}^{\mathrm{vis},\mathcal{L}}. \qquad (29)$$

Let $z_{\mathrm{LV},k} \triangleq \tilde{\boldsymbol{v}}_k^{\mathrm{vis},\mathcal{L}} - \hat{\boldsymbol{R}}_k^{\top}\hat{\boldsymbol{v}}_k$, the error observation model of LV is established as

$$z_{\mathrm{LV},k} = \boldsymbol{H}_{\mathrm{LV},k}\boldsymbol{\delta}_{x,k} + \boldsymbol{w}_k^{\mathrm{vis},\mathcal{L}} \qquad (30)$$

where

$$\boldsymbol{H}_{\mathrm{LV},k} = \begin{bmatrix}\boldsymbol{0}_3 & \hat{\boldsymbol{R}}_k^{\top}[\hat{\boldsymbol{v}}_k]_{\times} & \hat{\boldsymbol{R}}_k^{\top} & \boldsymbol{0}_3 & \boldsymbol{0}_3\end{bmatrix}.$$

TABLE II

FOUR FUSION MODES AND THEIR CONFIGURATIONS

Fusion Mode	EAS Update	VIS Update	LS Update
Mode 0	PAI	—	—
Mode 1	GVV	PAI	—
Mode 2	GVV	—	PAI
Mode 3	GVV	LV	PAI

E. Update Using Subsystem Outputs

When the observation information arrives, the estimator will automatically call the corresponding error observation model to update the state according to the current fusion mode. The detailed configurations of four fusion modes are listed in Table II, where each fusion mode represents a way in which subsystem variables participate in fusion. For example, in Mode 3, the estimator fuses GVV variable of EAS, LV variable of VIS, and PAI variable of LS using GVV, LV, and PAI error observation models, respectively.

1) Error State Update: For the convenience of representation, the error observation models of PAI, GVV, and LV are uniformly expressed as

$$z_k = \boldsymbol{H}_k\boldsymbol{\delta}_{x,k} + \boldsymbol{w}_{\mathrm{meas},k} \qquad (31)$$

where, z_k is error observation value; $\boldsymbol{H}_k$ is error observation matrix; $\boldsymbol{w}_{\mathrm{meas},k} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma}_{\mathrm{meas},k})$ is observation noise. The error state update is completed as follows:

$$\begin{aligned}\boldsymbol{S}_k &= \boldsymbol{H}_k\boldsymbol{P}_{k|k-1}\boldsymbol{H}_k^{\top} + \boldsymbol{\Sigma}_{\mathrm{meas},k} && \text{(32a)}\\ \boldsymbol{K}_k &= \boldsymbol{P}_{k|k-1}\boldsymbol{H}_k^{\top}\boldsymbol{S}_k^{-1} && \text{(32b)}\\ \hat{\boldsymbol{\delta}}_{x,k|k} &= \hat{\boldsymbol{\delta}}_{x,k|k-1} + \boldsymbol{K}_k(z_k - \boldsymbol{H}_k\hat{\boldsymbol{\delta}}_{x,k|k-1}) && \text{(32c)}\\ \boldsymbol{P}_{k|k} &= (\boldsymbol{1}_{15} - \boldsymbol{K}_k\boldsymbol{H}_k)\boldsymbol{P}_{k|k-1}(\boldsymbol{1}_{15} - \boldsymbol{K}_k\boldsymbol{H}_k)^{\top} && \text{(32d)}\\ &\quad + \boldsymbol{K}_k\boldsymbol{\Sigma}_{\mathrm{meas},k}\boldsymbol{K}_k^{\top}. && \text{(32e)}\end{aligned}$$

2) Full State Update: After obtaining the error posterior estimate, $\hat{\boldsymbol{\delta}}_{x,k|k}$, it will be used to compensate the predicted value of the full state, $\hat{\boldsymbol{x}}_{k|k-1}$, according to the inverse formula of (17). The full state update is abbreviated as

$$\hat{\boldsymbol{x}}_{k|k} = \hat{\boldsymbol{x}}_{k|k-1} \oplus \hat{\boldsymbol{\delta}}_{x,k|k}. \qquad (33)$$

The detailed calculation equation for each variable is as follows:

$$\begin{aligned}\hat{\boldsymbol{p}}_{k|k} &= \hat{\boldsymbol{p}}_{k|k-1} + \hat{\boldsymbol{\delta}}_{p,k|k} && \text{(34a)}\\ \hat{\boldsymbol{R}}_{k|k} &= \exp([\hat{\boldsymbol{\delta}}_{a,k|k}]_{\times})\hat{\boldsymbol{R}}_{k|k-1} && \text{(34b)}\\ \hat{\boldsymbol{v}}_{k|k} &= \hat{\boldsymbol{v}}_{k|k-1} + \hat{\boldsymbol{\delta}}_{v,k|k} && \text{(34c)}\\ \hat{\boldsymbol{b}}_{f,k|k} &= \hat{\boldsymbol{b}}_{f,k|k-1} + \hat{\boldsymbol{\delta}}_{bf,k|k} && \text{(34d)}\\ \hat{\boldsymbol{b}}_{\omega,k|k} &= \hat{\boldsymbol{b}}_{\omega,k|k-1} + \hat{\boldsymbol{\delta}}_{b\omega,k|k}. && \text{(34e)}\end{aligned}$$

It should be noted that, once the error compensation is completed, $\hat{\boldsymbol{\delta}}_{x,k|k}$ needs to be zeroed, and this operation is noted as

$$\hat{\boldsymbol{\delta}}_{x,k|k} \leftarrow \boldsymbol{0}_{15\times 1}. \qquad (35)$$

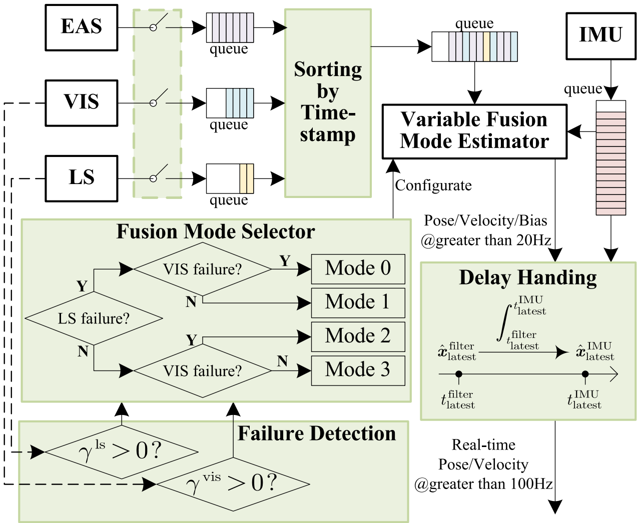
EAS
VIS
LS
queue
queue
queue
Sorting by Time-stamp
queue
IMU
queue
Variable Fusion Mode Estimator
Configurate
Pose/Velocity/Bias
@greater than 20Hz
Fusion Mode Selector
LS failure?
Y
N
VIS failure?
Y
N
Mode 0
Mode 1
VIS failure?
Y
N
Mode 2
Mode 3
Delay Handing
$\int_{t^{\text{filter}}_{\text{latest}}}^{t^{\text{IMU}}_{\text{latest}}}$
$\hat{\boldsymbol{x}}^{\text{filter}}_{\text{latest}}$
$\hat{\boldsymbol{x}}^{\text{IMU}}_{\text{latest}}$
$t^{\text{filter}}_{\text{latest}}$
$t^{\text{IMU}}_{\text{latest}}$
Failure Detection
$\gamma^{\text{ls}} > 0?$
$\gamma^{\text{vis}} > 0?$
Real-time
Pose/Velocity
@greater than 100Hz

Fig. 4. Flowchart of information manager.

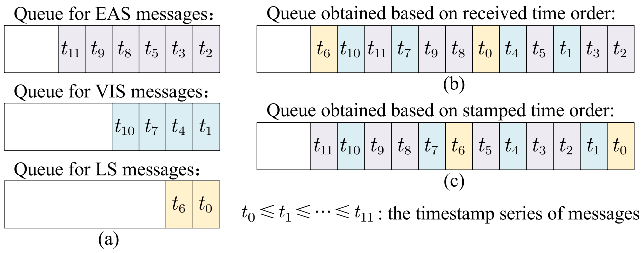
Queue for EAS messages:
t_{11} t_9 t_8 t_5 t_3 t_2
Queue for VIS messages:
t_{10} t_7 t_4 t_1
Queue for LS messages:
t_6 t_0
(a)
Queue obtained based on received time order:
t_6 t_{10} t_{11} t_7 t_9 t_8 t_0 t_4 t_5 t_1 t_3 t_2
(b)
Queue obtained based on stamped time order:
t_{11} t_{10} t_9 t_8 t_7 t_6 t_5 t_4 t_3 t_2 t_1 t_0
(c)
$t_0 \leqslant t_1 \leqslant \cdots \leqslant t_{11}$: the timestamp series of messages

Fig. 5. Example demonstrating the delay and disorder problem. (a) Queues for messages output by subsystems. (b) Queue obtained by sorting according to the time order of fusion center receiving them (it only shows one possible case). (c) Queue obtained by sorting according to the stamped order of the messages themselves.

VI. Information Manager

Fig. 4 presents the flowchart of our information manager. The FD module is designed to detect the significant degeneration of vision and laser subsystems. The fusion mode of the estimator is dynamically adjusted by *fusion mode selector* (FMS) module according to the detection results of FD, so as to isolate invalid observation information. The *sorting by timestamp* and *delay handling* module are designed to solve disorder information and delay problem.

A. Sorting Messages by Timestamp and Delay Handling

Because three subsystems have very different output frequency and processing delay, if those messages are not sorted correctly, the time order of fusion center receiving messages is usually inconsistent with the timestamp order of the messages themselves. Fig. 5 vividly demonstrates a example of disorder information problem. Obviously, the timestamp order shown in Fig. 5(c) is expected to be used by our fusion algorithm. Our solution to overcome this problem is: firstly, sorting them according to the timestamps of messages; then, using them to finish filtering update state in chronological order; finally, integrating inertial data to obtain real-time pose estimates. The detailed steps are as follows.

S1 *(Queue Checking)*: If there is at least one data in every subsystem buffer, entering S2; otherwise, waiting.

S2 *(Timestamp Sorting):* Sorting them according to their timestamps and putting them into a queue for filtering update.

S3 *(State Update):* Using data from update queue to update the state in the chronological order; the latest time, position estimate and attitude estimate of estimator are recorded as $t^{\text{filter}}_{\text{latest}}$, $\hat{\boldsymbol{p}}^{\text{filter}}_{\text{latest}}$ and $\hat{\boldsymbol{R}}^{\text{filter}}_{\text{latest}}$, respectively.

S4 *(Delay Compensating):* Let the latest time of obtained inertial data be $t^{\text{IMU}}_{\text{latest}}$. Integrating inertial data from $t^{\text{filter}}_{\text{latest}}$ to $t^{\text{IMU}}_{\text{latest}}$ to get real-time pose estimates, $\hat{\boldsymbol{p}}^{\text{IMU}}_{\text{latest}}$, $\hat{\boldsymbol{R}}^{\text{IMU}}_{\text{latest}}$, using full state prediction equation (21).

S5 *(Real-Time Outputting):* Outputting real-time estimate, and returning to S1.

B. FD and Handling

1) VIS Failure Detection: Aggression motion, texture-less scene etc., may cause the nonlinear optimization in VIS to fail. When the system fails, some quantities in modules of VIS often show anomalies, such as insufficient number of tracked features and large estimate of accelerometer or gyroscope bias. Therefore, we choose the average value of the number of tracked features, IMU bias, PI and AI in 1 s as degeneration metrics of VIS, denoted as $\bar{N}_f$, $\bar{\boldsymbol{b}}^{\text{vis}}_f$, $\bar{\boldsymbol{b}}^{\text{vis}}_\omega$, $\bar{\boldsymbol{\Delta}}^{\text{vis}}_p$ and $\bar{\boldsymbol{\Delta}}^{\text{vis}}_R$, respectively. Based on the above analysis, the discriminant of health status of VIS is designed as

$$\gamma^{\text{vis}} = \begin{cases} & \bar{N}_f < \alpha_N \text{ or } \left\|\bar{\boldsymbol{b}}^{\text{vis}}_f\right\| > \alpha_{bf} \\ 1, & \text{if or } \left\|\bar{\boldsymbol{b}}^{\text{vis}}_\omega\right\| > \alpha_{b\omega} \text{ or } \left\|\bar{\boldsymbol{\Delta}}^{\text{vis}}_p\right\| > \alpha_{\Delta_p} \\ & \text{or } \left\|\operatorname{Log}\left(\bar{\boldsymbol{\Delta}}^{\text{vis}}_R\right)\right\| > \alpha_{\Delta_R} \\ 0, & \text{otherwise} \end{cases} \tag{36}$$

here, $\gamma^{\text{vis}} = 1, 0$ represents failure and normal status of VIS, respectively. $\alpha_N, \alpha_{bf}, \alpha_{b\omega}, \alpha_{\Delta_p}, \alpha_{\Delta_R}$ are the thresholds, whose values are set to 5, 2, 0.02, 0.25, 0.07 in our experiments.

2) LS Failure Detection: We adapt the eigenvalue-analysis method [30] to detect failure of LS caused by encountering degraded scenes where scan-to-submap problem will be ill-constrained. According to Gauss–Newton or Levenberg–Marquardt algorithm [43] for nonlinear least-squares problems, in each iteration process, the original minimization problem i.e., (14) will be linearized at the working point, and the incremental equation of the following form will be obtained

$$\boldsymbol{H}\Delta\boldsymbol{x} = \boldsymbol{m} \tag{37}$$

where, symmetric $\boldsymbol{H} \in \mathbb{R}^{6\times6}$ and $\boldsymbol{m} \in \mathbb{R}^6$ are obtained by linearization at working point. $\Delta\boldsymbol{x} \in \mathbb{R}^6$ is the increment of variables used in the actual optimization process, and in this article we choose $\Delta\boldsymbol{x} = [\Delta p_x, \Delta p_y, \Delta p_z, \Delta\theta_x, \Delta\theta_y, \Delta\theta_z]^\top$. According to Lemma 2 in [30], the minimum eigenvalue of $\boldsymbol{H}$ is selected as the degradation metric of LS in this article. Once it is less than the given threshold, the *FD* module considers LS is severely degraded and the outputs of LS should not enter the *fusion center* to participate in data fusion. The discriminant of

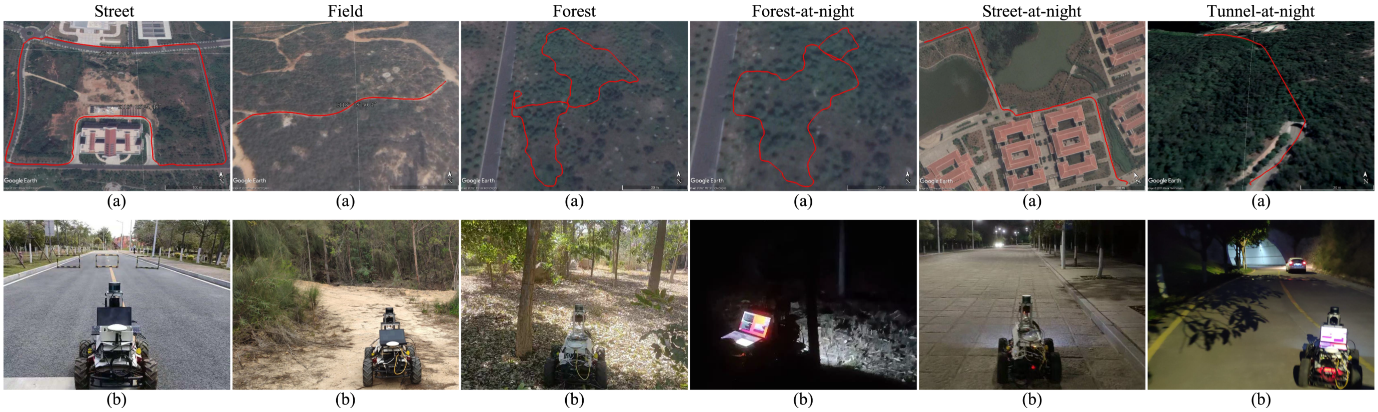

Fig. 6. Six test scenes. For each scene, the red line in (a) represents the motion trajectory of the vehicle in Google Earth and (b) photograph at the starting point. Note: at night, the forward LED light will be turned on.

health status of LS is designed as

$$\gamma^{\text{ls}} = \begin{cases} 1, & \text{if } \bar{\lambda}_{\min} < \alpha_\lambda \\ 0, & \text{if } \bar{\lambda}_{\min} \geq \alpha_\lambda \end{cases} \tag{38}$$

here, $\gamma^{\text{ls}} = 1, 0$ represents failure and normal status of LS, respectively. $\bar{\lambda}_{\min}$ is the average value of the minimum eigenvalue of $\boldsymbol{H}$ in 1 s, and α_λ is the given threshold, whose value is set to 130 for us.

3) Failure Handing: If subsystem VIS or LS fails (i.e., $\gamma^{\text{vis}} = 1$ or $\gamma^{\text{ls}} = 1$), it can be considered that the corresponding subsystem has a serious degradation problem. Next, FMS selects a new fusion mode for the estimator according to the fusion mode configuration table (see Table II) that does not use the failed subsystems. The process of online selecting fusion mode is vividly shown in the FMS module in Fig. 4.

VII. Experimental Results

A. Experimental Setup and Datasets

Fig. 1 presents our setup for data collecting, which consists of a vehicle base, multiple sensors and computer. In the figure, we list the technical details about the system hardware, down to specific component names and some of their high-level specifications. M600mini-G RTK system connect to RTK Networks infrastructure using mobile phone wireless networks to listen to GNSS correction information and use an antenna to receive GNSS observation information. Based on these two kinds of information, it provides the RTK positioning solution (abbreviated as RTK data), whose positioning accuracy is about 10 mm + 1 ppm for the horizontal plane, and 20 mm + 1 ppm for the vertical direction. XSENS MTi-G-710 GNSS/INS system, based on GNSS single point positioning solution and IMU, provides the integrated positioning solution (abbreviated as GNSS data), whose horizontal positioning accuracy is about 2.5 m circular error probability (CEP).

Extensive evaluations of the system are conducted using our six self-gathered dataset, named respectively: street, field, forest, forest-at-night, street-at-night, and tunnel-at-night. The reason for using our self-gathered datasets is that the current popular datasets, such as KITTI odometry datasets [44] and UrbanLoco datasets [25], do not contain encoder or wheel speedometer data and are not suitable for our investigation. Fig. 6 shows the motion trajectory and the photograph at the starting point for each of five scenes. The overall motion information of our vehicle in each scene is listed in Table III. In the first five scenes, RTK data is used as the ground truth. In the sixth scene where the length of the tunnel is about 120 m, GNSS data is used as the reference outside the tunnel as RTK system cannot work normally whether inside or outside the tunnel.

TABLE III
Overall Motion Information of Our Vehicle in Each Scene

Scene	Total time [s]	Total distance [m]	Reference
Street	1180	1460	RTK Data
Field	208	175	RTK Data
Forest	660	380	RTK Data
Forest-at-night	393	242	RTK Data
Street-at-night	542	666	RTK Data
Tunnel-at-night	256	305	GNSS Data

B. Evaluation Metrics

In order to quantify the positioning accuracy of the system, it is necessary to define appropriate metrics for quantitative analysis. Suppose that during the experiment, time series are $t_i \in \mathbb{R}, i = 1, 2, \ldots, N$, where $N \in \mathbb{N}^+$ is the total number of time series. The estimate of the position at each time is denoted as $\hat{\boldsymbol{p}}_i \in \mathbb{R}^3, i = 1, 2, \ldots, N$, the position reference is denoted as $\bar{\boldsymbol{p}}_i \in \mathbb{R}^3, i = 1, 2, \ldots, N$, and the traversed distance reference is recorded as $D_i \in \mathbb{R}, i = 1, 2, \ldots, N$. In the quantitative analysis, the ATE, ATE_i, is used to evaluate the absolute positioning accuracy at time t_i [45]. The relative trajectory error (RTE), RTE_i, is used to evaluate the positioning accuracy relative to traversed distance D_i at time t_i. Their mathematical definitions are

$$\text{ATE}_i = \|\hat{\boldsymbol{p}}_i - \bar{\boldsymbol{p}}_i\| \tag{39a}$$

$$\text{RTE}_i = \frac{\text{ATE}_i}{D_i} \times 100\%. \tag{39b}$$

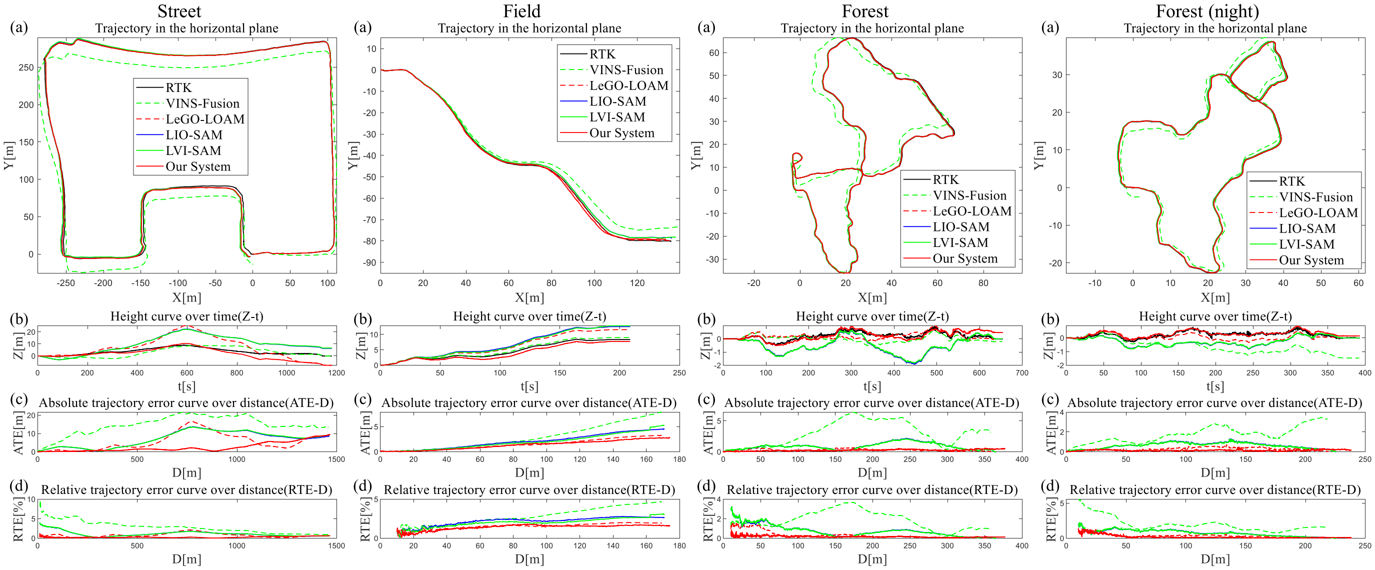

Fig. 7. Test results of first four scenes. For each scene, (a) trajectories in the horizontal plane, (b) curves of height over time, (c) curves of ATE over distance, and (d) curves of RTE over distance (starting at 10 m).

TABLE IV

MODAL COMBINATION OF ALGORITHMS TESTED IN THE EXPERIMENT

Algorithm	LiDAR	IMU	Camera	Encoder	AHRS
VINS-Fusion [33]	–	✓	✓	–	–
LeGO-LOAM [31]	✓	–	–	–	–
LIO-SAM [32]	✓	✓	–	–	–
LVI-SAM [12]	✓	✓	✓	–	–
Our System	✓	✓	✓	✓	✓

C. Accuracy and CPU Usage Evaluation: Tests With Street, Field, Forest, and Forest-at-Night Datasets

1) Accuracy Evaluation: In the experiments, we compare the proposed system against a variety of the publicly available state-of-the-art odometry systems, selected to cover the range of modal combinations, as shown in Table IV. Although we want to compare against R^{3}LIVE [23], LOCUS [27] and LIC-Fusion 2.0 [21], the former is currently only available for solid-state LiDAR and the latter two are not open-source implementation.

During the whole test process of each dataset, all modalities can work normally, there is no failure of VIS or LS. Fig. 7 shows the positioning trajectories and error curves in the first four scenarios. From the figure it can be found that: 1) the horizontal positioning trajectories of the algorithms using the laser modality are very close, which may be due to the fact that they all use the LOAM-based point cloud matching method; 2) the height change curve of our system is closer to RTK than the other algorithms; and 3) the change curve of ATE and RTE of our system is almost always closer to the time axis.

The quantitative analyses are summarized in Table V, where *mean*, *std*, *max*, *min* represent the average value, standard deviation, maximum and minimum value of investigated data, respectively. Observing and comparing the mean and std value of ATE and RTE, it can be seen that in the first four scenes, the positioning error of our system is the smallest. It is fully demonstrated that when VIS and LS work normally, our system has better positioning accuracy than the other systems.

2) CPU Usage Evaluation: The consumption of computing resources for system operation is also an important indicator for evaluating its performance and engineering feasibility. To this end, we performed a statistical analysis of the CPU loads in the first 200 s of the forest dataset. The CPU usage here refers to the number of CPU cores occupied by the algorithm when the algorithm runs on an desktop computer with Intel i7-8700K CPU @ 3.70 GHz × 12 and 16 RAM. The quantitative statistics are listed in the last four columns of Table V. The box plot drawn based on the original CPU loads data is shown in Fig. 8. The quantitative comparison of CPU usage shows that the computational consumption of our system is higher than that of VINS-Fusion, LeGO-LOAM and LIO-SAM, but about 13% less that LVI-SAM.

D. Robustness Evaluation: Test With Street-at-Night Dataset

In the street-at-night scene, the vehicle walks out of a "w"-shaped trajectory, whose height change about 5.4 m. There are relatively more street lights in the first half, and fewer street lights in the second half. On this dataset, we test various algorithms listed in Table IV, and the positioning trajectory and error curves of each algorithm are shown in Fig. 9. During testing we found that both our VIS and VINS-Fusion fail (Ceres solver prompts that an infinite value appears in the process of solving the nonlinear optimization problem). In order to further verify the effectiveness and feasibility of our FD method, we also tested the proposed system with the FD function turned off, denoted as Our System (no FD), and its results are also shown in Fig. 9. From the quantitative analysis results of each algorithm listed in Table VI, it can be seen that the mean value of ATE and RTE of Our System is comparable to that of LIO-SAM and LVI-SAM, and our std value is the smallest. Comparing the quantitative results of

TABLE V

QUANTITATIVE COMPARISON OF FIRST FOUR DATASETS USING VARIOUS ALGORITHMS

Algorithm	Street				Field				Forest				Forest-at-night				CPU Usage*			
	ATE [m]		RTE [%]		ATE [m]		RTE [%]		ATE [m]		RTE [%]		ATE [m]		RTE [%]		Num. of Cores			
	mean	std	mean	std	mean	std	mean	std	mean	std	mean	std	mean	std	mean	std	mean	std	max	min
VINS-Fusion	14.81	4.90	2.71	1.46	3.40	2.68	3.08	1.12	2.85	2.03	1.86	0.94	1.74	0.88	2.00	0.98	1.45	0.13	1.72	1.04
LeGO-LOAM	5.80	4.57	0.77	0.52	1.64	1.10	1.60	0.36	0.28	0.18	0.33	0.40	0.27	0.14	0.35	**0.20**	**0.72**	**0.07**	**0.88**	**0.51**
LIO-SAM	7.09	4.09	1.09	0.66	2.32	1.53	2.24	0.53	0.82	0.57	0.72	0.59	0.57	0.33	0.74	0.45	1.48	0.15	2.11	1.08
LVI-SAM	7.19	4.10	1.10	0.66	2.26	1.66	2.13	0.61	0.81	0.56	0.72	0.60	0.56	0.32	0.74	0.46	3.27	0.45	4.01	1.50
Our System	**2.78**	**2.82**	**0.30**	**0.16**	**1.49**	**0.93**	**1.43**	**0.33**	**0.24**	**0.14**	**0.16**	**0.07**	**0.14**	**0.06**	**0.22**	0.24	2.85	0.26	3.32	2.12

*Usage of CPU is computed from the forest dataset.

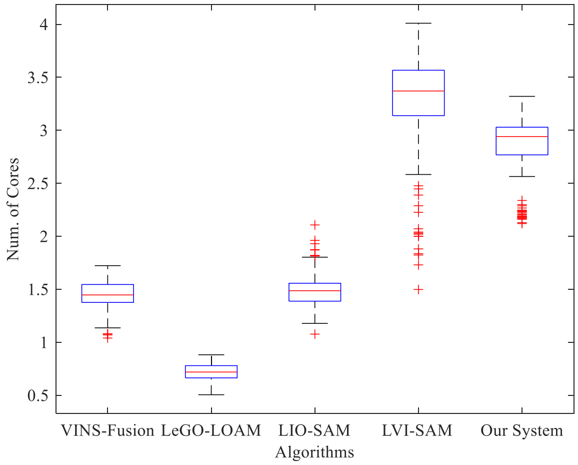

Fig. 8. Boxplot visualization of the CPU usage computed for the different algorithms in the first 200 s of the forest dataset.

TABLE VI

QUANTITATIVE COMPARISON OF STREET-AT-NIGHT DATASET USING VARIOUS ALGORITHMS

Algorithm	ATE [m]		RTE [%]	
	mean	std	mean	std
VINS-Fusion	519.60	878.49	95.94	142.64
LeGO-LOAM	2.45	2.54	0.60	0.36
LIO-SAM	**1.05**	1.01	**0.32**	0.21
LVI-SAM	1.67	1.82	0.46	0.30
Our System	1.12	**0.87**	0.45	**0.18**
Our System (no FD)	6.83	10.93	1.55	1.74

Our System and Our System (no FD), it can be found that FD and FMS can effectively detect and handle VIS failure, and it also shows that timely isolation of failed visual information sources is necessary to improve system performance.

In order to more clearly show the FD and handling process, during the test, we collect the degeneration metrics of VIS and LS, $\bar{N}_f$, $\|\bar{\boldsymbol{b}}_f^{\text{vis}}\|$, $\|\bar{\boldsymbol{b}}_\omega^{\text{vis}}\|$, $\|\bar{\boldsymbol{\Delta}}_p^{\text{vis}}\|$, $\|\text{Log}(\bar{\boldsymbol{\Delta}}_R^{\text{vis}})\|$, $\bar{\lambda}_{\min}$, and the health status of VIS and LS, γ^{vis}, γ^{ls}, as well as the fusion mode of the estimator, and plot them in Fig. 10. According to the health status discriminant of VIS and LS, (36) and (38), Fig. 10(a)–(e) tell us that the VIS begin to fail at about 300 s and Fig. 10(f) tells us that LS don't fail during the whole movement. It can be seen from Fig. 10(g)–(i) that FD can effectively detect the failure of VIS, and the FMS correctly switches the fusion mode of the estimator from modes 3 to 2 according to the FD results of FD.

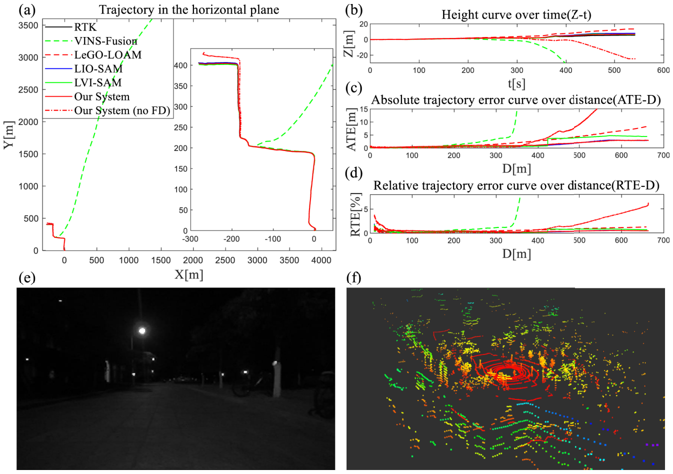

Fig. 9. Test results of street-at-night scene. (a) Trajectories in the horizontal plane. (b) Curves of height over time. (c) Curves of ATE over distance. (d) Curves of RTE over distance (starting at 10 m). (e) and (f) Gray image from left camera and a point cloud from LiDAR, respectively, when VIS fails.

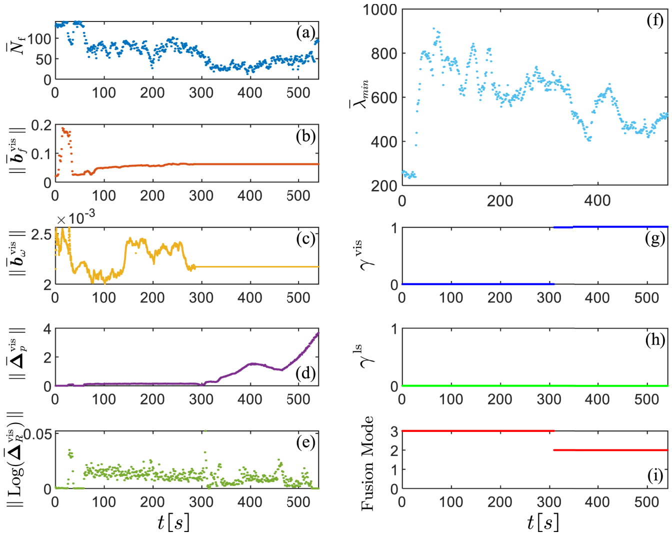

Fig. 10. Inside the FD and FMS module, variables change with time. (a)–(e) Time-varying curves of VIS degeneracy metrics described in (36). (f) Time-varying curve of LS degeneracy metric described in (38). (g) and (h) Change of the health status of VIS and LS, respectively, here 1 and 0, respectively, represents failure and normal status. (i) Fusion mode change process of the estimator.

E. Robustness Evaluation: Test With Tunnel-at-Night Dataset

In the tunnel-at-night scene, our vehicle passed through a tunnel about 120 m. Because the environmental geometry

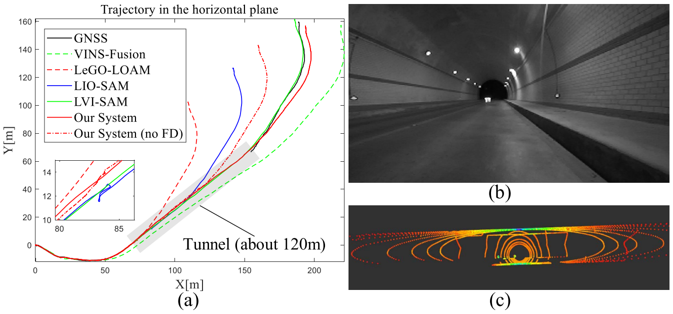

Fig. 11. Test results for tunnel scene. (a) Trajectories of different systems in horizontal plane. Here, FD stands for FD introduced in Section VI-B. (b) and (c) Gray photograph camera and the point cloud from LiDAR, respectively, when laser modality occurs degeneracy. GNSS data is selected as reference.

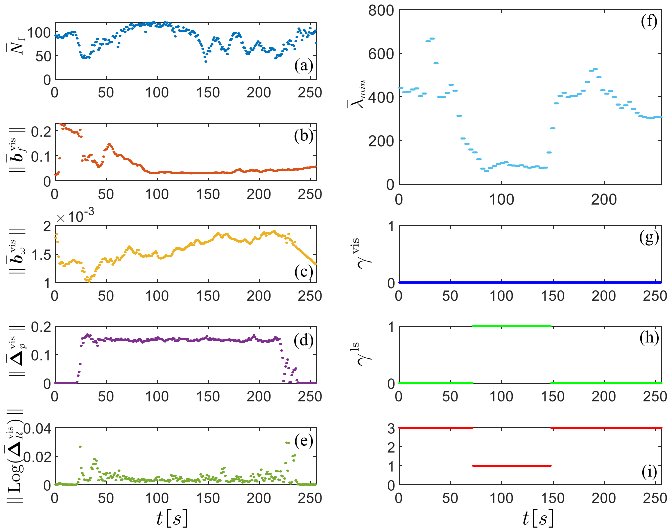

Fig. 12. Inside the information manager module, each variable changes with time. (a)–(e) show the time-varying curves of VIS degeneracy metrics described in (36). (f) Time-varying curve of LS degeneracy metric described in (38). (g) and (h) Change of the health status of VIS and LS, respectively, here 1 and 0, respectively, represents failure and normal status. (i) Fusion mode change process of the estimator.

of tunnel is too single, the laser modality degenerates and LS fails. Algorithms listed in Table IV and Our System (no FD) are tested on this dataset. The positioning trajectories of algorithms, an image and a point cloud in the tunnel are shown in Fig. 11. It can be seen from Fig. 11(a) that, the positioning accuracy of our system is second only to LVI-SAM, and is significantly better than the other algorithms. If the GNSS data is used as a reference at the endpoint, the horizontal positioning error of VINS-Fusion, LeGO-LOAM, LIO-SAM, LVI-SAM, Our System, Our System (no FD) are about 30.6 m (10.2%), 97.7 m (32.6%), 57.3 m (19.1%), 3.2 m (1.1%), 5.7 m (1.9%) and 33.3 m (11.1%), respectively. It should be noted that due to the multipath effects and nonline-of-sight [13] caused by the occlusion or reflection of the mountain near the tunnel, the GNSS data is likely to have systematic errors, so the quantitative results here are likely to deviate from the real situation to a certain extent.

The change curves of the variables related to FD and FMS during the test are plotted in Fig. 12. According to failure discriminant of VIS and LS, (36) and (38), Fig. 12(a)–(e) tell us that the VIS does not fail and Fig. 12(f) tells us that LS fails when the vehicle moves in the tunnel. It can be seen from Fig. 12(g)–(i) that FD can effectively detect the failure of LS, and the FMS correctly switches the fusion mode of the estimator between modes 3 and 1 according to the FD results of FD.

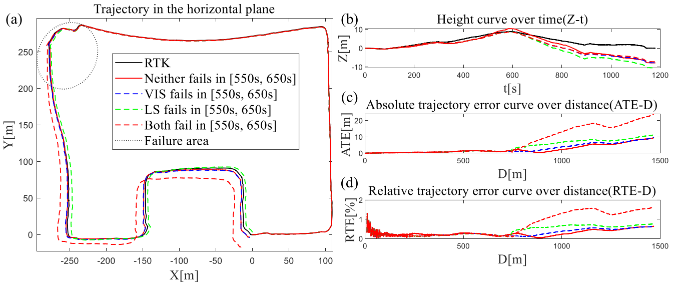

Fig. 13. Test results on the semi-simulated street dataset. (a) Trajectories in the horizontal plane. (b) Curves of height over time. (c) Curves of ATE over distance. (d) Curves of RTE over distance (starting at 10 m). RTK data is selected as reference.

TABLE VII

Quantitative Analysis Results for Semi-Simulated Street Dataset

Failure situation	ATE [m]		RTE [%]		Fusion mode change		
in [550s, 650s]	mean	std	mean	std	befere	[550s, 650s]	after
Neither fails	2.78	2.82	0.30	0.16	3	3	3
VIS fails	3.12	2.99	0.33	0.17	3	2	3
LS fails	4.22	3.73	0.44	0.24	3	1	3
Both fail	8.20	8.40	0.77	0.60	3	0	3

F. Robustness Evaluation: Test With Semi-Simulated Street Dataset

In the street-at-night and tunnel-at-night test above, the performance of the system is tested only when LS fails or only when VIS fails, and there is a lack of testing for failure of both VIS and LS. In order to further verify the robustness of the system to more complex failure situations, semi-simulation experiments are carried out on the street dataset. By artificially setting the health status of VIS and LS, we create three failure cases during [550, 650] s in street dataset: 1) only VIS fails; 2) only LS fails; and 3) both VIS and LS fail, and then test our system for each situation. The estimated trajectories, height and error change curves of each algorithm are shown in Fig. 13. The quantitative analysis results listed in Table VII show that our system can effectively detect and handle the failure of VIS or LS. For example, when both VIS and LS fail, FMS will set the estimator's fusion mode to 0, and when both VIS and LS return to normal, FMS will restore the estimator's fusion mode to 3. Despite the failure of 100 s, the system can still provide continuous, relatively reliable output, and the ATE and RTE will not change by order of magnitude.

VIII. Conclusion and Future Work

This article proposed a low-drift and high-robustness state estimation system for improving the persistence and reliability of UGV positioning in challenging environment, which integrates 3-D LiDAR, IMUs, stereo camera, encoders, and AHRS. It can effectively detect and isolate failed subsystems, and use the other ones to achieve good accuracy as much as possible. It has been evaluated in six scenarios: street, field, forest, forest-at-night, street-at-night and tunnel-at-night: 1) when both VIS and LS work normally, the overall RTE of the system is about 0.5%; 2) in the scene where the vision subsystem fails, the system can continue to provide positioning information with about 0.5% accuracy level almost unaffected; 3) in the tunnel scene with serious laser degeneracy, it can still maintain RTE about 1.9%; and 4) even if both VIS and LS fail for a short period of time, the relative positioning error does not change by orders of magnitude.

Compared with the existing approaches, our system has the following significant advantages: 1) the proposed architecture has generality and engineering convenience, i.e., its subsystems can easily be replaced with any available odometry or localization algorithm; 2) our system fuses more sensing modalities, and even when vision and laser modalities are serious degraded, the system can still rely on IMU and encoder to provide relative reliable positioning in a short period of time; 3) our system is more computationally efficient, e.g., its CPU usage is reduced about 13% compared to LVI-SAM using IMU, vision and laser modalities; and 4) the scenes for testing is more rich, including street, forest, field, tunnel, day and night.

Although the system has good positioning accuracy and robustness in typical scenes, there is still room for improvement: 1) further integrate GNSS positioning information when it is available, which is conducive to eliminate cumulative errors in large-scale autonomous movement and 2) in addition to pursuing positioning performance, UGV autonomous navigation also hopes to build maps simultaneously to provide an environmental model for its path planning and obstacle avoidance planning. Therefore, our future research will focus on these issues and create a resilient navigation system that is independent of navigation satellites.

References

[1] R. G. Arrshith, K. S. Suhas, C. Tejas, and G. Subramaniyam, "Unmanned ground vehicle (UGV)—Defense bot," in *Proc. 2nd Int. Conf. Inventive Syst. Control (ICISC)*, Jan. 2018, pp. 1201–1205.

[2] J. Ma, M. Bajracharya, S. Susca, L. Matthies, and M. Malchano, "Real-time pose estimation of a dynamic quadruped in GPS-denied environments for 24-hour operation," *Int. J. Robot. Res.*, vol. 35, no. 6, pp. 631–653, May 2016.

[3] S. H. Young, T. A. Mazzuchi, and S. Sarkani, "A framework for predicting future system performance in autonomous unmanned ground vehicles," *IEEE Trans. Syst. Man Cybern. Syst.*, vol. 47, no. 7, pp. 1192–1206, Jul. 2017.

[4] G. Wan *et al.*, "Robust and precise vehicle localization based on multi-sensor fusion in diverse city scenes," in *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, May 2018, pp. 4670–4677.

[5] M. Nezhadshahbodaghi and M. R. Mosavi, "A loosely-coupled EMD-denoised stereo VO/INS/GPS integration system in GNSS-denied environments," *Measurement*, vol. 183, Oct. 2021, Art. no. 109895.

[6] E. Jones and S. Soatto, "Visual-inertial navigation, mapping and localization: A scalable real-time causal approach," *Int. J. Robot. Res.*, vol. 30, no. 4, pp. 407–430, 2011.

[7] W. J. Lv, Y. Kang, and J. H. Qin, "Indoor localization for skid-steering mobile robot by fusing encoder, gyroscope, and magnetometer," *IEEE Trans. Syst. Man Cybern. Syst.*, vol. 49, no. 6, pp. 1241–1253, Jun. 2019.

[8] X. Bai, W. Wen, and L.-T. Hsu, "Degeneration-aware outlier mitigation for visual inertial integrated navigation system in urban canyons," *IEEE Trans. Instrum. Meas.*, vol. 70, pp. 1–15, 2021.

[9] C. Li, "Multi-sensor fusion for robust simultaneous localization and mapping," M.S. thesis, Robot. Inst., Carnegie Mellon Univ., Pittsburgh, PA, USA, Aug. 2019.

[10] T. Du, S. Shi, Y. Zeng, J. Yang, and L. Guo, "An integrated INS/LiDAR odometry/polarized camera pose estimation via factor graph optimization for sparse environment," *IEEE Trans. Instrum. Meas.*, vol. 71, pp. 1–11, 2022.

[11] X. Li and Q. Xu, "A reliable fusion positioning strategy for land vehicles in GPS-denied environments based on low-cost sensors," *IEEE Trans. Ind. Electron.*, vol. 64, no. 4, pp. 3205–3215, Apr. 2017.

[12] T. Shan, B. Englot, C. Ratti, and D. Rus, "LVI-SAM: Tightly-coupled LiDAR-Visual-Inertial odometry via smoothing and mapping," in *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*. Xi'an, China, May 2021, pp. 5692–5698.

[13] W. Wen, T. Pfeifer, X. Bai, and L.-T. Hsu, "It is time for factor graph optimization for GNSS/INS integration: Comparison between FGO and EKF," 2020, *arXiv:2004.10572*.

[14] M. Brossard, S. Bonnabel, and A. Barrau, "Unscented Kalman filter on lie groups for visual inertial odometry," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2018, pp. 649–655.

[15] X. Lyu, B. Hu, Z. Wang, D. Gao, K. Li, and L. Chang, "A SINS/GNSS/VDM integrated navigation fault-tolerant mechanism based on adaptive information sharing factor," *IEEE Trans. Instrum. Meas.*, vol. 71, pp. 1–13, 2022.

[16] 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.

[17] G. He, X. Yuan, Y. Zhuang, and H. Hu, "An integrated GNSS/LiDAR-SLAM pose estimation framework for large-scale map building in partially GNSS-denied environments," *IEEE Trans. Instrum. Meas.*, vol. 70, pp. 1–9, 2021.

[18] V. Madyastha, V. Ravindra, S. Mallikarjunan, and A. Goyal, "Extended Kalman filter vs. error state Kalman filter for aircraft attitude estimation," in *Proc. AIAA Guid., Navigat., Control Conf.*, 2011, p. 6615.

[19] J. Solà, "Quaternion kinematics for the error-state Kalman filter," 2017, *arXiv:1711.02508*.

[20] X. Zuo, P. Geneva, W. Lee, Y. Liu, and G. Huang, "LIC-fusion: LiDAR-Inertial-Camera odometry," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Jul. 2019, pp. 5848–5854.

[21] X. Zuo *et al.*, "LIC-fusion 2.0: LiDAR-Inertial-Camera odometry with sliding-window plane-feature tracking," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Jan. 2021, pp. 5112–5119.

[22] J. Lin, C. Zheng, W. Xu, and F. Zhang, "R^{2}LIVE: A robust, real-time, LiDAR-inertial-visual tightly-coupled state estimator and mapping," *IEEE Robot. Autom. Lett.*, vol. 6, no. 4, pp. 7469–7476, Oct. 2021.

[23] J. Lin and F. Zhang, "R3LIVE: A robust, real-time, RGB-colored, LiDAR-inertial-visual tightly-coupled state estimation and mapping package," 2021, *arXiv:2109.07982*.

[24] J. Zhang and S. Singh, "Laser-visual-inertial odometry and mapping with high robustness and low drift," *J. Field Robot.*, vol. 35, no. 8, pp. 1242–1264, Dec. 2018.

[25] W. Wen *et al.*, "UrbanLoco: A full sensor suite dataset for mapping and localization in urban scenes," in *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, Aug. 2020, pp. 2310–2316.

[26] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The KITTI dataset," *Int. J. Robot. Res.*, vol. 32, no. 11, pp. 1231–1237, 2013.

[27] M. Palieri *et al.*, "LOCUS: A multi-sensor LiDAR-centric solution for high-precision odometry and 3D mapping in real-time," *IEEE Robot. Autom. Lett.*, vol. 6, no. 2, pp. 421–428, Apr. 2021.

[28] V. Kubelka, L. Oswald, F. Pomerleau, F. Colas, T. Svoboda, and M. Reinstein, "Robust data fusion of multimodal sensory information for mobile robots," *J. Field Robot.*, vol. 32, no. 4, pp. 447–473, 2015.

[29] J. Simanek, V. Kubelka, and M. Reinstein, "Improving multi-modal data fusion by anomaly detection," *Auto. Robots*, vol. 39, no. 2, pp. 139–154, 2015.

[30] J. Zhang, M. Kaess, and S. Singh, "On degeneracy of optimization-based state estimation problems," in *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*. Piscataway, NJ, USA, May 2016, pp. 809–816.

[31] T. Shan and B. Englot, "LeGO-LOAM: Lightweight and ground-optimized LiDAR odometry and mapping on variable terrain," in *Proc. Int. Conf. Intell. Robots Syst.*, Oct. 2018, pp. 4758–4765.

[32] T. Shan, B. Englot, D. Meyers, W. Wang, C. Ratti, and D. Rus, “LIO-SAM: Tightly-coupled LiDAR inertial odometry via smoothing and mapping,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Jan. 2021, pp. 5135–5142.

[33] T. Qin, J. Pan, S. Cao, and S. Shen, “A general optimization-based framework for local odometry estimation with multiple sensors,” 2019, *arXiv:1901.03638*.

[34] J. Zhang and S. Singh, “Visual-LiDAR odometry and mapping: Low-drift, robust, and fast,” in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2015, pp. 2174–2181.

[35] S. Rabiee and J. Biswas, “A friction-based kinematic model for skid-steer wheeled mobile robots,” in *Proc. Int. Conf. Robot. Automat. IEEE Int. Conf. Robot. Automat. (ICRA)*. New York, NY, USA, May 2019, pp. 8563–8569.

[36] M. T. Sabet, H. M. Daniali, A. Fathi, and E. Alizadeh, “A low-cost dead reckoning navigation system for an AUV using a robust AHRS: Design and experimental analysis,” *IEEE J. Ocean. Eng.*, vol. 43, no. 4, pp. 927–939, Oct. 2018.

[37] J. Shi and C. Tomasi, “Good features to track,” in *Proc. IEEE (CVPR)*, Jun. 1994, pp. 593–600.

[38] B. D. Lucas and T. Kanade, “An iterative image registration technique with an application to stereo vision,” in *Proc. 7th Int. J. Conf. Artif. Intell.*, vol. 2. Vancouver, BC, Canada, Aug. 1981, pp. 674–679.

[39] C. Forster, L. Carlone, F. Dellaert, and D. Scaramuzza, “On-manifold preintegration for real-time visual–inertial odometry,” *IEEE Trans. Robot.*, vol. 33, no. 1, pp. 1–21, Feb. 2017.

[40] M. Himmelsbach, F. V. Hundelshausen, and H.-J. Wuensche, “Fast segmentation of 3D point clouds for ground vehicles,” in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2010, pp. 560–565.

[41] I. Bogoslavskyi and C. Stachniss, “Fast range image-based segmentation of sparse 3D laser scans for online operation,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*. Daejeon, South Korea, Oct. 2016, pp. 163–169.

[42] J. Zhang and S. Singh, “Low-drift and real-time LiDAR odometry and mapping,” *Auton. Robots*, vol. 41, no. 2, pp. 401–416, Feb. 2017.

[43] J. Nocedal and S. J. Wright, *Numerical Optimization* (Springer Series in Operation Research and Financial Engineering), 2nd ed. New York, NY, USA: Springer, 2006.

[44] A. Geiger, P. Lenz, and R. Urtasun, “Are we ready for autonomous driving? The KITTI vision benchmark suite,” in *Proc. CVPR*, Sep. 2012, pp. 3354–3361.

[45] Z. Zhang and D. Scaramuzza, “A tutorial on quantitative trajectory evaluation for visual(-inertial) odometry,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*. Madrid, Spain, Oct. 2018, pp. 7244–7251.

Dongjie Wu received the B.S. degree in energy application from Jimei University, Xiamen, China, in 2015, and the M.S. degree in automation from Xiamen University, Xiamen, in 2019, where he is currently pursuing the Ph.D. degree with the Department of Automation.

His research interests include data fusion, autonomous navigation, machine learning, and visual servoing.

Xunyu Zhong received the M.E. degree in mechatronics engineering and the Ph.D. degree in control theory and control engineering from Harbin Engineering University, Harbin, Heilongjiang, China, in 2007 and 2009, respectively.

He is currently an Associate Professor with the Department of Automation, Xiamen University, Xiamen, China. His current research interests include autonomous robotics (perception, localization, and planning), computer vision, and artificial intelligence.

Xiafu Peng received the M.S. degree in control theory and applications from the Harbin Institute of Naval Engineering, Harbin, China, in 1994, and the Ph.D. degree in control science and engineering from Harbin Engineering University, Harbin, in 2001.

Since 2002, he has been with the Department of Automation, Xiamen University, Xiamen, China, where he is currently a Professor. His research interests include inertial technology and intelligent transportation systems.

Huosheng Hu (Life Senior Member, IEEE) received the M.Sc. degree in industrial automation from Central South University, Changsha, China, in 1982, and the Ph.D. degree in robotics from the University of Oxford, Oxford, U.K., in 1993.

He is currently a Professor with the School of Computer Science and Electronic Engineering, University of Essex, Colchester, U.K., where he is leading the Robotics Research Group. He has authored or coauthored more than 420 articles. His current research interests include robotics, human–robot interaction, embedded systems, mechatronics, and pervasive computing.

Prof. Hu is a Founding Member of the IEEE Robotics and Automation Society Technical Committee on Networked Robots, a fellow of the Institution of Engineering and Technology, and a Senior Member of the Association for Computing Machinery. He also serves as the Editor-in-Chief of the *International Journal of Automation and Computing* and the online *Robotics* journal and an Executive Editor of the *International Journal of Mechatronics and Automation*.

Qiang Liu received the B.Eng. and M.Eng. degrees in automation and control engineering from the Harbin Institute of Technology, Harbin, China, in 2012 and 2014, respectively, and the Ph.D. degree in computer science from the University of Essex, Colchester, U.K., in 2019.

He is currently a Post-Doctoral Researcher with the Department of Psychiatry, University of Oxford, Oxford, U.K. His research interests include machine learning algorithms and deep neural networks; statistical analysis; medical data analysis; image classification, segmentation, and clustering; biomedical signal processing; smart sensors, wearable sensors, sensor integration, and data fusion algorithms; visual simultaneous localization and mapping (SLAM) and scene understanding; and natural language processing (NLP) algorithms.