2024 IEEE International Conference on Robotics and Automation (ICRA)
May 13-17, 2024. Yokohama, Japan

2024 IEEE International Conference on Robotics and Automation (ICRA) | 979-8-3503-8457-4/24/$31.00 ©2024 IEEE | DOI: 10.1109/ICRA57147.2024.10610070

Ground-Fusion: A Low-cost Ground SLAM System Robust to Corner Cases

Jie Yin[1], Ang Li[1], Wei Xi[2], Wenxian Yu[1], and Danping Zou[1]*

***Abstract*— We introduce Ground-Fusion, a low-cost sensor fusion simultaneous localization and mapping (SLAM) system for ground vehicles. Our system features efficient initialization, effective sensor anomaly detection and handling, real-time dense color mapping, and robust localization in diverse environments. We tightly integrate RGB-D images, inertial measurements, wheel odometer and GNSS signals within a factor graph to achieve accurate and reliable localization both indoors and outdoors. To ensure successful initialization, we propose an efficient strategy that comprises three different methods: stationary, visual, and dynamic, tailored to handle diverse cases. Furthermore, we develop mechanisms to detect sensor anomalies and degradation, handling them adeptly to maintain system accuracy. Our experimental results on both public and self-collected datasets demonstrate that Ground-Fusion outperforms existing low-cost SLAM systems in corner cases. We release the code and datasets at https://github.com/SJTU-ViSYS/Ground-Fusion.**

I. INTRODUCTION

Ground robots find extensive applications in logistics, catering, and industrial production. In many scenarios, it is critical to reliably navigate the ground robots within both indoor [1] and outdoor environments [2]. Simultaneous Localization and Mapping (SLAM) technology plays a key role in robot navigation. While LiDAR-based SLAM systems excel in many scenarios, their high costs are not suitable for low-cost applications where SLAM systems using affordable sensors such as Visual-Inertial Odometry (VIO) are preferred. However, VIO may exhibit reduced accuracy in specific motion modes that introduce unobserved degrees of freedom (DoFs), as discussed in [3]. To tackle this challenge, VINS-RGBD [4] integrated RGBD images with inertial data to avoid scale issues and achieve better pose estimation. Similarly, VINS-on-Wheels [5] introduced a wheel odometer-assisted VIO system to maintain a consistent metric scale. [6] [7] further loosely integrates GNSS signals into a visual-inertial-wheel-odometry. Additionally, [8] tightly combined GNSS raw measurements with the VIO system for drift-free global state estimation. [9] harnessed a sky-pointing camera for NLOS mitigation, thereby enhancing localization in urban canyons.

Nevertheless, our previous investigations [10] have revealed that the robustness of existing SLAM systems in challenging scenarios needs to be further improved. To address this concern, this paper focuses on two aspects: robust system initialization and corner case addressing. To ensure a successful initialization, we propose an effective strategy that includes three different methods: stationary, visual, and dynamic, which are designed to handle various situations. In addition, we discuss the possible sensor faults that might happen in corner cases [11] and handle them accordingly. We have conducted extensive experiments to evaluate our method. The results show the robustness of our method in different scenarios. We highlight the main contributions of this work as follows:

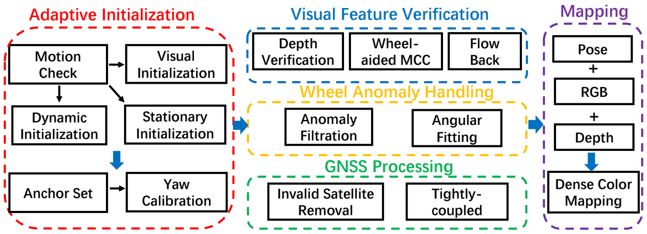


Fig. 1. The system adopts an adaptive initialization strategy based on the robot's motion state. Potential sensor faults will be detected and handled accordingly. Real-time dense color mapping is supported to facilitate navigation tasks.

- We implement a low-cost SLAM system that tightly couples GNSS-RGBD-IMU-Wheel sensors, which can work reliably both indoors and outdoors by fully exploiting each sensor to enable robust initialization in diverse cases.
- We propose effective strategies to detect and handle sensor faults that may arise in sensor fusion systems, including visual failures, wheel anomalies, and GNSS degradation, thereby greatly enhancing robustness.
- We present a SLAM dataset serving as a new benchmark for challenging corner cases.

II. RELATED WORK

A. Initialization of a multi-sensor SLAM system

Multi-sensor SLAM systems, particularly those that are tightly-coupled, are heavily reliant on high-quality initialization due to its profound impact on system robustness and accuracy. VINS-Mono [12] employs a hybrid initialization strategy that performs vision-only Structure from Motion (SfM) and a subsequent visual-IMU alignment. Then the scale and gravity direction will be further optimized. This method requires sufficient IMU excitation and visual parallax for successful initialization. However, the movement of a ground robot can easily experience unobservable DoFs,

[1] Shanghai Jiao Tong University. [2] Midea Corparate Research Center. * Corresponding Author: Danping Zou (dpzou@sjtu.edu.cn).

This work was supported by National Key R&D Program (2022YFB3903802), NSFC(62073214), and Midea Group's 3D Vision Project.

such as uniform motion, which makes the VIO systems unobservable at scale. To mitigate this issue, [4] utilizes depth information as the source of the scale information for initialization. Similarly, [13] uses wheel odometer measurements to refine the gravity vector, utilizing visual SfM for scale estimation. Moreover, [14] proposes a Zero Velocity Update (ZUPT) aided initialization method with stationary and dynamic phase to achieve better robustness. However, the initialization of these systems is still not robust enough to severe sensor faults. Building upon these insights, we introduce an adaptive initialization strategy encompassing three distinct approaches tailored to different scenarios in this work.

B. *Corner cases*

Visual challenge: We classify vision failures into three categories. The first type is insufficient-features problem, typically caused by lack of textures or inadequate light. This can lead to significant drift in visual systems [12], [15]. The second is characterized by no valid feature points due to significant occlusion or aggressive motion. This can result in unsuccessful system initialization or tracking failure as demonstrated in [11]. The third type refers to dynamic environments with numerous moving objects. The moving points on these objects can greatly degrade localization accuracy. Currently, there exist some effective semantic-based methods [16] [17] that well tackle this challenge. In this study, we focus on cost-efficient geometric approaches that do not utilize GPUs. Specifically, we reject dynamic features through feature filtration and depth validation.

Wheel odometer challenge: There are two types of wheel odometer challenges: inaccurate angular velocity and wheel anomaly. To tackle the former issue, [18] integrated measurements from wheel encoders and gyroscopes to formulate a relative motion constraint. For the latter challenge, they detect the wheel anomaly with visual information: If more than half of the previous features are considered as outliers during visual-wheel optimization, wheel slippage is detected. In response, they reset the current frame's initial state and eliminate current wheel measurements to mitigate the anomaly. However, a significant decrease in feature points could also arise from environmental changes, rather than just wheel-related factors. Therefore, this strategy is likely to mistakenly eliminate reliable wheel odometer observations. Considering this limitation, [19] introduced three techniques for actively detecting abnormal chassis movements with the help of motion constrains and IMU measurements. We integrate insights from these efforts to enhance our system's capacity to address wheel odometer challenges.

GNSS degradation: Typically, there are three kinds of GNSS challenges: low-speed movements, less than four satellites, and no GNSS signals. Early work employed Receiver Autonomous Integrity Monitoring (RAIM) scheme [20] to assess integrity performance levels of GNSS systems, detecting and mitigating errors within the GNSS receiver based on residuals. As a GNSS-Visual-IMU tightly-coupled system, GVINS [8] proposes strategies to address GNSS-related issues in challenging scenarios. GVINS [8] remains robust in less-than-four-satellite case due to its tighly-coupled framework. In GNSS-denied environments, GVINS degenerates into a convectional VIO. Building upon these insights, our system firstly filters out unreliable satellites with threshold-based methods. Furthermore, our system monitors low-speed states and ensures that GNSS-related factors are not optimized in these scenarios.

IMU saturation: He et al. [21] address IMU saturation issues for drones with highly aggressive motions, while this situation is rarely encountered by safety-critical ground robots. In our study, we consider short-term IMU observations within a sliding window to be reliable and free from faults.

C. *Ground robot datasets*

A high-quality dataset can stimulate progress in SLAM. While there have been a few datasets captured by ground robots [22] [23] [24], many of them are outdated and not challenging enough to current advanced SLAM systems. Our previous work [10] presents a challenging dataset M2DGR [1], which has been met with great interest from SLAM researchers. Therefore, we further extend the M2DGR dataset by adding extra low-cost sensors, including a wheel encoder, a 2D LiDAR and a depth camera. This not only allows us to test our proposed system, but also launches a novel benchmark for future research on multi-sensor SLAM.

III. Methodology

Our Ground-Fusion system tightly integrates RGB images, depth information, inertial information, and wheel odometer measurements within the optimization framework. All the sensory measurements are maintained in a sliding window for real-time performance. The system consists of adaptive initialization, a multi-sensor state estimator with corner case addressing, and a dense mapping module. Before introducing each module, we clarify the notations and frames used in this paper: $(\cdot)^w$ denotes the world frame. $(\cdot)^b$ is the body (IMU) frame, and $(\cdot)^c$ is the camera frame. $\mathbf{q}_b^w$ and $\mathbf{p}_b^w$ represent the rotation and translation from the body frame to the world frame, respectively. b_j and c_j are the body frame and the camera frame when capturing the jth image.

A. *Adaptive initialization*

Our initialization module consists of three alternative strategies for local odometry initialization: dynamic method, visual method and stationary one. The system determines whether there is sufficient motion excitation based on the GLRT (Generalized Likelihood Ratio Test) [25] method, which is formulated as:

$$\mathbb{G} = \frac{1}{m}\sum_{t\in\psi}\left(\frac{1}{\sigma_a^2}\left\|\tilde{a}_t - g\frac{\bar{a}_t}{\|\bar{a}_t\|}\right\|^2 + \frac{1}{\sigma_\omega^2}\|\tilde{\omega}_t\|^2\right) \tag{1}$$

where $\{\tilde{a}_t, \tilde{\omega}_t\}$ are the raw measurements of the IMU, m is IMU measurement number within the sliding window, ψ denotes the window range, and $\bar{a}$ is the average acceleration.

[1] https://github.com/SJTU-ViSYS/M2DGR

The GLRT value roughly divides the motion state into the following three categories, where the threshold values for β and γ are determined by experimental approach:

$$\text{State} \approx \begin{cases} \text{Stationary} & (\mathbb{G} < \beta) \\ \text{Slow Motion} & (\beta \leq \mathbb{G} \leq \gamma) \\ \text{Aggressive Motion} & (\mathbb{G} > \gamma) \end{cases} \tag{2}$$

Next, we'll further validate the motion state and describe the corresponding initialization method for each kind of motion.

Stationary: If the $\mathbb{G}$ is below the value β, we introduce wheel and visual observations to further ensure whether the system is static. We utilize wheel median integration method to predict the pose:

$$\begin{aligned} \mathbf{R}_{j+1}^{o} &= \mathbf{R}_{j}^{o} \operatorname{Exp}\left(\overline{\boldsymbol{\omega}}_{j+1}^{o} \Delta t\right) \\ \boldsymbol{p}_{j+1}^{w} &= \boldsymbol{p}_{j+1}^{w} + \overline{\boldsymbol{v}}_{j+1}^{w} \Delta t \end{aligned} \tag{3}$$

where Δt is the time difference between odometer frames o_j and o_{j+1}, and $\overline{\boldsymbol{\omega}}_{j+1}^{o} = \frac{1}{2}\left(\boldsymbol{\omega}_{j}^{o} + \tilde{\boldsymbol{\omega}}_{j+1}^{o}\right)$ and $\overline{\boldsymbol{v}}_{j+1}^{w} = \frac{1}{2}\left(\boldsymbol{v}_{j}^{w} + \tilde{\boldsymbol{v}}_{j+1}^{w}\right)$. Assuming there are n odometer frames between consecutive images c_k and c_{k+1}, the wheel preintegration pose's norm between them can be expressed as:

$$\mathbb{W} = \left\| \boldsymbol{p}_{j+n}^{w} - \boldsymbol{p}_{j}^{w} \right\|^{2} \tag{4}$$

Additionaly, we can extract feature points from the latest frame match them with images in the sliding window. The average visual parallax can then be formulated as:

$$\mathbb{V} = \frac{1}{m} \sum_{i \in [0, m-1]} \left(\sum_{j \in [0, r-1]} \left\| \boldsymbol{p}_{i}^{j} - \boldsymbol{p}_{m-1}^{j} \right\|^{2} \right) \tag{5}$$

where r is the matched feature point number between jth image and the latest image.

In the initialization phase, if at least two of the stationary criteria $\{\mathbb{G} < \beta, \mathbb{W} < \eta, \mathbb{V} < \theta\}$ are met (all threshold values determined by experimental approach), we consider the vehicle is static. Otherwise, we treat the vehicle as in motion and use the methods in the next paragraph for initialization. In the confirmed stationary case, we establish the first camera frame as the local world frame and align the z-axis with the gravity direction. Subsequently, all other poses within the sliding window are aligned with the first pose, while the velocity is set to zero. The system state $\boldsymbol{p}, \boldsymbol{v}, \boldsymbol{q}$ will be set to a constant block during optimization. The stationary detection and ZUPT is not only applicable to the initialization phase, but also used throughout the optimization process.

Slow motion: In the slow motion case, the camera pose $\left(\boldsymbol{p}_{c}^{w}, \boldsymbol{q}_{c_k}^{w}\right)$ between the two frames could be computed by solving a PnP (Perspective-n-Point) problem. Since the RGB-D camera can directly measure the depth information, the IMU pose can be calculated without scale parameter by:

$$\begin{aligned} q_{b_k}^{w} &= q_{c_k}^{w} \otimes \left(q_{c_k}^{b_k}\right)^{-1} \\ p_{b_k}^{w} &= p_{c_k}^{w} - R_{b_k}^{w} p_{c_k}^{b_k} \end{aligned} \tag{6}$$

where the extrinsic $\left(\boldsymbol{p}_{c}^{b}, \boldsymbol{q}_{c}^{b}\right)$ is provided offline.

Combining above states with the IMU pre-integration term γ, we can calibrate the gyroscope bias by minimizing following least-square function:

$$\begin{aligned} \min\nolimits_{\delta b_w} \textstyle\sum_{k \in B} \left\| {q_{b_{k+1}}^{c_0}}^{-1} \otimes q_{b_k}^{c_0} \otimes \gamma_{b_{k+1}}^{b_k} \right\|^{2} \\ \gamma_{b_{k+1}}^{b_k} \approx \hat{\gamma}_{b_{k+1}}^{b_k} \otimes \begin{bmatrix} 1 \\ \frac{1}{2} J_{b_w}^{\gamma} \delta b_w \end{bmatrix} \end{aligned} \tag{7}$$

where B represents all frame indexes in the window Since the scale is known, later parameters initialization only contain velocities and gravity vector.

$$X_I = \left[\boldsymbol{v}_{b_0}^{b_0}, \boldsymbol{v}_{b_1}^{b_1}, \cdots, \boldsymbol{v}_{b_n}^{b_n}, g^{C_0}\right] \tag{8}$$

Considering two consecutive IMU frames b_k and b_{k+1}, we have following equations:

$$\begin{aligned} \alpha_{b_{k+1}}^{b_k} &= \mathbf{R}_{c_0}^{b_k} \left(\overline{\boldsymbol{p}}_{b_{k+1}}^{C_0} - \overline{\boldsymbol{p}}_{b_k}^{c_0} - \mathbf{R}_{b_k}^{c_0} \boldsymbol{v}_{b_k}^{b_k} \Delta t + \frac{1}{2} \boldsymbol{g}^{c_0} \Delta t^2 \right) \\ \beta_{b_{k+1}}^{b_k} &= \mathbf{R}_{c_0}^{b_k} \left(\mathbf{R}_{b_{k+1}}^{c_0} \boldsymbol{v}_{b_{k+1}}^{b_{k+1}} - \mathbf{R}_{b_k}^{c_0} \boldsymbol{v}_{b_k}^{b_k} + \boldsymbol{g}^{c_0} \Delta t \right) \end{aligned} \tag{9}$$

Combining equation (6) and equation (9), we can solve the initial values of X_I. Finally, the gravity vector obtained from the previous linear initialization step is further refined.

Aggressive motion: In highly aggressive motion, the visual features may be unstable due to motion blur or few overlap, causing the visual SfM-based initialization to be unreliable. By contrast, the wheel odometer measurement makes the velocity and scale observable, the pose can be calculated by wheel integration. Consequently, we opt not to employ visual SfM for pose estimation as described in Equation (6), but instead, we employ a wheel-aided initialization method. The camera pose in equation (9) could be replaced by wheel odometer pose to solve the gyroscope bias. To establish a consistent reference frame, we define the world frame using the first wheel frame, aligning its z-axis with that of the wheel frame. In comparison to the approach used in [13], which exclusively employs the wheel odometer for scale refinement, our method eliminates the redundant SfM component, fully harnessing the wheel odometer for a more efficient initialization process. It's worth noting that while this initialization method does not rely on visual information, once successfully initialized, the visual factor can still be integrated into the tightly coupled optimization process when the system identifies effective feature points. After successful local initialization by any of above three methods, we perform a three-step global initialization, which are adapted from [8].

B. Multi-sensor state estimator with corner case handling

We formulate the state estimation as a maximizing a posteriori (MAP) problem. We follow the factor graph framework of [5] which maintains a sliding window, and further extend to a GNSS-RGBD-IMU-Wheel fusion system. The calculation of residuals and Jacobi can refer to the previous literature [8] [4] [5]. Next, we mainly introduce how our system processes sensor measurements to become more robust to corner cases.

Wheel anomalies: The wheel odometer measurement can be formulated as $\{\tilde{v}_t^o = v_t^o + n_t^v, \tilde{\omega}_t^o = \omega_t^o + n_t^{\omega}\}$. Here $\{\tilde{v}_t, \tilde{\omega}_t\}$ are the raw measurements of the wheel odometer,

$v^o = \left[\begin{array}{ccc} v_x^o & v_y^o & 0 \end{array}\right]^T, \omega^o = \left[\begin{array}{ccc} 0 & 0 & \omega_z^o \end{array}\right]^T$ denote the velocity and angular velocity in the wheel odometer frame

The error of wheel odometer mainly comes from inaccurate angular velocity estimation and sudden chassis anomaly, such as wheel slippage and collision. Since the IMU's angular velocity measurement is more reliable and has a higher frame rate than the wheel odometer, we replace the original wheel odometer measurement with the IMU angular velocity using a linear fitting method:

$$\omega_z^o = \mathbf{R}_b^o\left(\omega_m^b + \frac{\omega_n^b - \omega_m^b}{t_n - t_m}(t - t_m) - \boldsymbol{b}_\omega\right)_z \tag{10}$$

where t_m and t_n are two closest IMU timestamps to current wheel measurement.

To detect wheel anomalies, we compare the pre-integration of both the IMU and the wheel odometer measurements between the current frame and the second latest frame. If the difference of their resulting poses' norm surpasses the threshold $\varepsilon = 0.015$, we see it as a wheel abnormality. In this case, we refrain from incorporating the current wheel odometer observations into subsequent optimization process.

Vision anomalies: Our system employs the KLT sparse optical flow algorithm [26] for tracking feature points as adapted from [12]. Three visual challenges include no-valid-feature problem during initialization, insufficient-feature issue during the localization and dynamic environments The first one has been solved in Sec III (a), while the second one can be mitigated by tightly-coupled wheel odometer and IMU data. To address dynamic objects, We further introduce two strategies: feature filtration and depth validation.

For feature filtration, we firstly adopt the flow back method by reversing the order of the two frames for optical flow backtracking. Only the feature points that are successfully tracked during both iterations and exhibit an adjacent distance below a specified threshold are retained for further process. Moreover, we introduce a wheel-assisted moving consistency check (MCC) approach. Our system utilizes the wheel-preintegration pose and previous optimized poses. For a feature observed for the first time in the i-th image and subsequently observed in other m images within the sliding windows, we define the average reprojection residual r_k of the feature observations as:

$$\boldsymbol{r}_k = \frac{1}{m}\sum_{j \neq i}\left\|\boldsymbol{u}_k^{c_i} - \pi\left(\mathbf{T}_b^c \mathbf{T}_w^{b_i} \mathbf{T}_{b_j}^w \mathbf{T}_c^b \mathbf{P}_k^{c_j}\right)\right\| \tag{11}$$

here $\mathbf{u}_k^{c_i}$ represents the observation of the k-th feature in the i-th frame, and $\mathbf{P}_k^{c_j}$ is the 3D location of the k-th feature in the j-th frame. The function π denotes the camera projection model. When the value of r_k exceeds a preset threshold, the k-th feature is considered dynamic and will be removed from further process. This approach offers the advantage of preemptively eliminating potential error-tracking features in current image frame before the optimization phase.

For the depth validation, we initiate by associating the depth measurements acquired from the depth camera with each pixel representing a feature point. In cases where the depth measurement surpasses the effective range of the depth camera, the pixel is temporarily left empty. Subsequently, we employ the triangulation method on the RGB image to calculate the depth of the feature points, thus filling in all the pixel depths. Additionally, for those feature points where the disparity between the depth measured by the depth camera and the depth computed through triangulation is below a predefined threshold, we record their indexes and fix their depth values to a constant value during the optimization phase.

GNSS anomalies: Three GNSS challenge scenarios include too-few-satellites, no-satellite-signal, and low-speed movement. In the first two cases, [8] has proven that with the help of the tightly-coupled GNSS-Visual-Inertial framework, limited reliable satellites can still be effective in improving the global state estimation, and the GVIO system will degrade into a VIO system when no GNSS signals observed. In this work, our system firstly filters out unreliable satellites with excessive pseudo-range and Doppler uncertainty, those with an insufficient number of tracking times, and those at a low elevation angle. In low-speed scenarios with a GNSS receiver velocity below the threshold $v_{ths} = 0.3m/s$ (the noise level of the Doppler shift), we do not involve GNSS factors in the optimization to prevent GNSS noise from corrupting the state estimation.

IV. Experiments

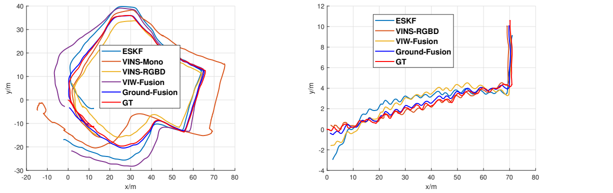
A. Benchmark tests

Localization performance: Openloris-Scene [24] is a SLAM dataset collected by a ground robot with an RGBD camera, an IMU and a wheel odometer. Ground-Fusion is tested against cutting-edge SLAM systems on three scenarios of Openloris-Scene [24], namely Office (7 seqs), home (5 seqs) and corridor (3 seqs). Table I shows that Ground-Fusion performs well on these scenarios.

Initialization performance: We conduct initialization tests on Ground-Challenge dataset [11] with complex sequences in corner cases[2]. To evaluate the efficiency of system initialization, we measure the time required for each system to complete the initialization process, which is defined as the difference between the timestamp of the first observation received by the system and that of the first output pose. In terms of the quality of initialization, we evaluate the Absolute Trajectory Error (ATE) RMSE [22] for each system, focusing on the initial 10 seconds of each sequence. We select several challenging sequences from Ground-Challenge [11] for the initialization test. These sequences include $Office3$, which features changing light conditions; $Darkroom2$ recorded in a dark room; and $Wall2$ captured in front of a textureless wall. Additionally, $Motionblur3$ exhibits severe camera motion blur during aggressive movement, while $Occlusion4$ involves severe camera occlusion. And $Static1$ was recorded in a stationary state with wheel suspension.

We evaluate our method against baseline methods on these challenging sequences. Results in Table II demonstrate that our method excels in both quality and efficiency of

[2] https://github.com/sjtuyinjie/Ground-Challenge



(a). Loop2 (b). Corridor1

Fig. 2. Estimated and ground-truth (GT) trajectories on part of sample sequences are visualized on the x-y plane.

initialization. Specifically, in Sequence $Office3$, our system adaptively chooses visual initialization method, requiring the least amount of time. In visual challenging scenarios with limited textures ($Darkroom2$ and $Wall2$) and visual failure($Motionblur3$ and $Occlusion4$), our system adopts the dynamic initialization method and initialize successfully with high efficiency, while some baseline systems fail due to inadequate feature points for visual SfM. In stationary case $Static1$, our system initializes successfully with the stationary method, while baseline methods fail due to insufficient motion excitation.

TABLE I
AVERAGE ATE RMSE(M) OF SLAM SYSTEMS ON CHALLENGING SEQUENCES OF OPENLORIS-SCENE [24] DATASET

	ORBSLAM3 [15]	DSO [27]	VINS-Mono [12]	InfiniT [28]	Elastic-F [29]	Ours
office	0.52	0.75	0.19	0.16	**0.13**	0.18
home	1.11	0.67	0.54	0.58	1.45	**0.46**
corridor	2.98	10.66	3.41	3.43	17.76	**0.71**

TABLE II
INITIALIZATION TIME COST (S) AND ATE RMSE(M) OF SLAM SYSTEMS ON SAMPLE SEQUENCES

Sequence	VINS-Mono [12]	VINS-RGBD [4]	VIW-Fusion	Ours
Office3	2.09 / **0.02**	2.10 / **0.02**	2.42 / **0.02**	**1.37** / **0.02**
Darkroom2	**1.30** / 2.06	2.52 / 0.04	1.61 / 0.04	1.36 / **0.03**
Wall2	2.50 / 0.08	**1.27** / 0.07	2.00 / 0.05	1.37 / **0.03**
Motionblur3	fail	fail	1.30 / 0.11	**1.30** / **0.09**
Occlusion4	fail	fail	fail	**5.52** / **0.14**
Static1	fail	fail	fail	**1.42** / **0.00**

TABLE III
ATE RMSE(M) OF SLAM SYSTEMS ON SAMPLE SEQUENCES

Sequence	VINS-Mono [12]	VINS-RGBD [4]	VIW-Fusion	TartanVO [30]	ESKF	Ours
Office3	0.34	0.31	0.18	1.52	0.15	**0.14**
Darkroom2	86.06	0.82	0.53	1.18	0.38	**0.22**
Wall2	1.21	1.00	0.15	2.76	0.20	**0.12**
Hall1	7.06	94.27	0.85	3.08	1.29	**0.36**
Rotation3	29.12	0.19	0.18	0.13	0.14	**0.08**
Motionblur3	9.37	32.31	0.78	1.61	0.44	**0.26**
Occlusion4	—	—	—	2.35	0.16	**0.15**
Roughroad3	0.17	25.52	0.14	0.41	0.17	**0.11**
Slope1	9.41	2.84	0.65	3.13	3.13	**0.64**
Loop2	6.09	3.44	9.23	13.18	6.31	**2.28**
Corridor1	4.48	0.85	1.12	2.05	1.55	**0.74**
Static1	—	—	—	**0.01**	2.87	**0.01**

Visual challenges: We further introduce two new sequences for evaluation: Sequence $Hall1$ was recorded in a highly dynamic hall with a lot of people moving around; Sequence $Rotation3$ involves the robot rotating without much translation, which can influence the triangulation process in the visual front-end. We evaluated cutting-edge SLAM systems along with our method on the aforementioned sequences. The evaluated algorithms include VINS-Mono [12], VINS-RGBD [4], VIW-Fusion [3] and TartanVO [30](a learning-based vo). Additionally, we implement an ESKF-based IMU-wheel fusion odometry as a baseline without visual input. The results are shown in Table III, and trajectories for some sample sequences are visualized in Figure 2. Overall, Ground-Fusion achieves the best localization results across all tested sequences.

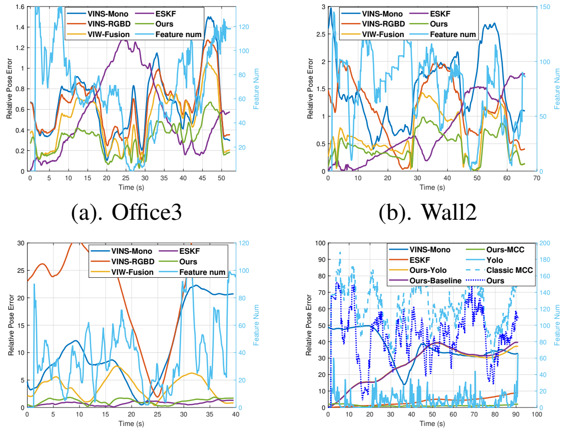


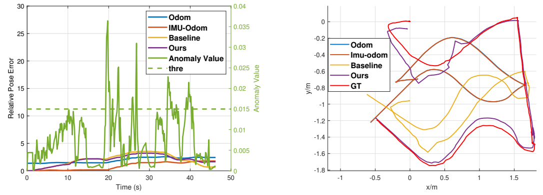
Fig. 3. The relative pose errors (m) of each method and the number of effective feature points over time on some visual challenging sequences are plotted.

To illustrate the visual challenges, we plot the relative pose error (RPE) of each method and the number of valid visual feature points over time in Figure 3. The results show that insufficient feature points greatly degrade the performance of the visual front-end. For instance, at 25 seconds in Sequence $Office3$, the feature point number suddenly drops to zero due to loss, causing a notable increase in RPE for VINS-Mono [12] and VINS-RGBD [4]. Similarly, VINS-Mono struggles to estimate the depth by triangulation during pure rotation ($Rotation3$), resulting in significant drift. In such scenarios, our system still performs well due to the tightly-coupled wheel odometer. In $Occlusion4$ with no valid feature points observed, most systems including VIW-Fusion, fail initialization. By contrast, our system initializes using a wheel-aided dynamic approach and outperforms the wheel-IMU fusion ESKF baseline in localization accuracy.

For the dynamic environment ($Hall1$), VINS-Mono [12] suffers from significant drift. We conducted a comparison between the method using a YOLO [31] module to remove features within the bounding box of pre-defined moving categories (e.g. people) and a baseline method with a classic MCC using the optimized pose. In Figure 3 (d), while YOLO significantly reduces the number of feature points, but some of them are not actually on dynamic objects. Consequently, the RPE does not show a significant decrease when compared to the baseline. When incorporating wheel-aided MCC method, the system effectively eliminates genuine dynamic points, leading to a significant improvement in positioning accuracy. To further validate the efficacy of wheel-aided MCC, we conducted ablation tests on all seven visual challenge sequences, showing an average decrease of **0.07**m in the ATE RMSE compared to the baseline method.

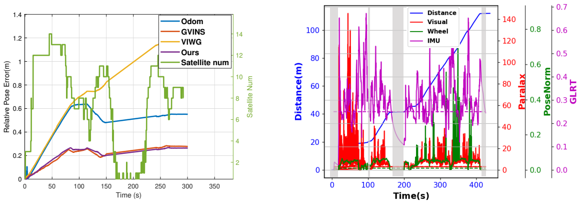
Wheel odometer challenges: In sequences $Roughroad3$

[3] https://github.com/TouchDeeper/VIW-Fusion



(a) (b)

Fig. 4. (a) Wheel anomaly analysis and (b) Trajectory of different methods in the *Anomaly* sequence.



(a) (b)

Fig. 5. (a) ATE RMSE(m) of SLAM systems on wheel anomaly sequences (b) The solid lines denote the value of each method, and dashed lines denote their corresponding thresholds. Grey shading denotes areas where at least two stationary conditions are satisfied.

and *Slope*1, the robot moves on a rough road and a steep slope respectively; In *Loop*2, the robot traverses a hall with a carpeted aisle loop, where wheels slip considerably; Sequence *Corridor*1 is a zigzag route in a long corridor. Table III shows that our method achieves the best performance in all these sequences. We further conducted ablation tests to verify the effect of IMU angular velocity as a substitute for wheel angular velocity. We selected two sequences with sharp turns, including *corridor*1 and *loop*2. The results in Table IV indicate that the IMU-odometer measurements contribute to a better localization accuracy.

TABLE IV

ATE RMSE(M) OF SLAM SYSTEMS ON SELECTED SEQUENCES

Seq.	Odom	IMU-Odom	Baseline	Ours
Corridor1	2.17	1.03	0.89	**0.66**
Loop2	17.60	6.53	4.66	**1.53**
Anomaly	0.78	0.77	0.62	**0.07**
Static	3.54	3.51	2.35	**0.01**

Moreover, we test our method on sequences with wheel anomaly. In the *Anomaly* sequence, the robot body moves as the carpet beneath it is pulled away, while the robot wheels do not move. Conversely, in the *Static* sequence the robot is suspended so the robot body will not move even when wheels are moving. The results of different methods on the two sequences are shown in Table IV, where ”baseline” denotes Ground-Fusion without wheel anomaly detection, while ”ours” denotes our full method. As depicted in Figure 4 (a), a wheel anomaly is evident between 20s and 40s. Our full method adeptly eliminates erroneous wheel odometer readings here. Figure 4 (b) shows that only our full method matches well with the ground-truth trajectory. Similarly, our method effectively detects the wheel anomaly in the *Static* sequence.

B. *Outdoor Tests*

We further evaluate our method in large-scale outdoor environments as follows. we built a ground robot for data collection, with all the sensors well-synchronized and calibrated. We recorded some sequences in various scenarios[4] and choose three most challenging sequences in this paper: In sequence *Lowspeed*, the ground vehicle moved at a low speed and made several stops; Sequence *Tree* was under dense tree cover, causing occlusion of the GNSS satellites; In sequence *Switch*, the vehicle transitioned from outdoors to indoors, and then returned outdoors again.

GNSS challenge: We evaluate our method under GNSS challenges against baseline methods, with their localization results shown in Table V. Overall, Ground-Fusion outperforms baseline methods in all these cases. In *Lowspeed*, when the robot is stationary, GVINS fails to localize due to Doppler noise, and the VINS-GW [5] also drifts, while Ground-Fusion works robustly by removing unreliable GNSS measurements. As shown in Figure 5(a), in *Switch*, VINS-GW experiences severe drift due to the loosely-coupled GNSS signals that deteriorate significantly as the robot approaches indoor areas, while both GVINS and our method remain unaffected due to their tightly-coupled integration. In Sequence *Tree*, GVINS falters due to unstable visual features, while our method remains robust due to tightly-coupled wheel and depth measurement.

TABLE V

ATE RMSE(M) OF SLAM SYSTEMS ON SAMPLE SEQUENCES

Sequence	Lowspeed	Switch	Tree
Raw Odom	8.88	4.95	2.88
SPP	2.54	6.60	3.03
GVINS [8]	fail	1.40	fail
VINS-RGBD [4]	4.72	1.70	2.27
VINS-GW	20.68	33.61	3.26
Ours	**0.63**	**1.32**	**0.55**

Zero velocity update (ZUPT): We visualize the three stationary detection methods with the GT distance on the *Lowspeed* sequence in Figure 5(b). The figure shows that a single sensor might misclassify the motion state. For instance, the wheel method fails to detect the stationary state at approximately 110 seconds. By contrast, our scheme combines three sensors, presenting reliable detection of the stationary state. Quantitatively, the ATE RMSE in *Lowspeed* decreased by **0.05m** after ZUPT.

V. CONCLUSION

This paper presents a tightly-coupled RGBD-Wheel-IMU-GNSS SLAM system to achieve reliable localization for ground vehicles. Our system features robust initialization through three strategies. We have also devised effective anomaly detection and handling methods to address corner cases, with experimental results demonstrating the superiority of our system.

[4] https://github.com/SJTU-ViSYS/M2DGR-plus
[5] https://github.com/Wallong/VINS-GPS-Wheel

References

[1] J. Yin, C. Liang, X. Li, Q. Xu, H. Wang, T. Fan, Z. Wu, and Z. Zhang, "Design, sensing and control of service robotic system for intelligent navigation and operation in internet data centers," in *2023 IEEE 19th International Conference on Automation Science and Engineering (CASE)*. IEEE, 2023, pp. 1–8.

[2] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and J. J. Leonard, "Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age," *IEEE Transactions on robotics*, vol. 32, no. 6, pp. 1309–1332, 2016.

[3] A. Martinelli, "Closed-form solution of visual-inertial structure from motion," *International Journal of Computer Vision*, vol. 106, pp. 138–152, 2013.

[4] Z. Shan, R. Li, and S. Schwertfeger, "Rgbd-inertial trajectory estimation and mapping for ground robots," *Sensors*, vol. 19, no. 10, p. 2251, 2019.

[5] K. J. Wu, C. X. Guo, G. Georgiou, and S. I. Roumeliotis, "Vins on wheels," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 5155–5162.

[6] T. Hua, L. Pei, T. Li, J. Yin, G. Liu, and W. Yu, "M2c-gvio: motion manifold constraint aided gnss-visual-inertial odometry for ground vehicles," *Satellite Navigation*, vol. 4, no. 1, pp. 1–15, 2023.

[7] J. Yin, H. Jiang, J. Wang, D. Yan, and H. Yin, "A robust and efficient ekf-based gnss-visual-inertial odometry," in *2023 IEEE International Conference on Robotics and Biomimetics (ROBIO)*. IEEE, 2023, pp. 1–5.

[8] S. Cao, X. Lu, and S. Shen, "Gvins: Tightly coupled gnss–visual–inertial fusion for smooth and consistent state estimation," *IEEE Transactions on Robotics*, 2022.

[9] J. Yin, T. Li, H. Yin, W. Yu, and D. Zou, "Sky-gvins: a sky-segmentation aided gnss-visual-inertial system for robust navigation in urban canyons," *Geo-spatial Information Science*, vol. 0, no. 0, pp. 1–11, 2023.

[10] J. Yin, A. Li, T. Li, W. Yu, and D. Zou, "M2dgr: A multi-sensor and multi-scenario slam dataset for ground robots," *IEEE Robotics and Automation Letters*, 2022.

[11] J. Yin, H. Yin, C. Liang, H. Jiang, and Z. Zhang, "Ground-challenge: A multi-sensor slam dataset focusing on corner cases for ground robots," in *2023 IEEE International Conference on Robotics and Biomimetics (ROBIO)*. IEEE, 2023, pp. 1–5.

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

[13] J. Liu, W. Gao, and Z. Hu, "Visual-inertial odometry tightly coupled with wheel encoder adopting robust initialization and online extrinsic calibration," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019, pp. 5391–5397.

[14] L. Gui, C. Zeng, S. Dauchert, J. Luo, and X. Wang, "A zupt aided initialization procedure for tightly-coupled lidar inertial odometry based slam system," *Journal of Intelligent & Robotic Systems*, vol. 108, no. 3, p. 40, 2023.

[15] C. Campos, R. Elvira, J. J. G. Rodríguez, J. M. Montiel, and J. D. Tardós, "Orb-slam3: An accurate open-source library for visual, visual–inertial, and multimap slam," *IEEE Transactions on Robotics*, 2021.

[16] J. Liu, X. Li, Y. Liu, and H. Chen, "Rgb-d inertial odometry for a resource-restricted robot in dynamic environments," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 9573–9580, 2022.

[17] C. Yu, Z. Liu, X.-J. Liu, F. Xie, Y. Yang, Q. Wei, and Q. Fei, "Ds-slam: A semantic visual slam towards dynamic environments," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 1168–1174.

[18] M. Quan, S. Piao, M. Tan, and S.-S. Huang, "Tightly-coupled monocular visual-odometric slam using wheels and a mems gyroscope," *IEEE Access*, vol. 7, pp. 97 374–97 389, 2019.

[19] G. Peng, Z. Lu, S. Chen, D. He, and L. Xinde, "Pose estimation based on wheel speed anomaly detection in monocular visual-inertial slam," *IEEE Sensors Journal*, vol. 21, no. 10, pp. 11 692–11 703, 2020.

[20] S. Hewitson and J. Wang, "Gnss receiver autonomous integrity monitoring (raim) performance analysis," *Gps Solutions*, vol. 10, pp. 155–170, 2006.

[21] D. He, W. Xu, N. Chen, F. Kong, C. Yuan, and F. Zhang, "Point-lio: Robust high-bandwidth light detection and ranging inertial odometry," *Advanced Intelligent Systems*, p. 2200459, 2023.

[22] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A benchmark for the evaluation of rgb-d slam systems," in *2012 IEEE/RSJ international conference on intelligent robots and systems (IROS)*. IEEE, 2012, pp. 573–580.

[23] K. Y. Leung, Y. Halpern, T. D. Barfoot, and H. H. Liu, "The utias multi-robot cooperative localization and mapping dataset," *The International Journal of Robotics Research*, vol. 30, no. 8, pp. 969–974, 2011.

[24] X. Shi, D. Li, P. Zhao, Q. Tian, Y. Tian, Q. Long, C. Zhu, J. Song, F. Qiao, L. Song, *et al.*, "Are we ready for service robots? the openloris-scene datasets for lifelong slam," in *2020 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2020, pp. 3139–3145.

[25] I. Skog, P. Handel, J.-O. Nilsson, and J. Rantakokko, "Zero-velocity detection—an algorithm evaluation," *IEEE transactions on biomedical engineering*, vol. 57, no. 11, pp. 2657–2666, 2010.

[26] B. D. Lucas, T. Kanade, *et al.*, *An iterative image registration technique with an application to stereo vision*. Vancouver, 1981, vol. 81.

[27] J. Engel, V. Koltun, and D. Cremers, "Direct sparse odometry," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 3, pp. 611–625, 2017.

[28] O. Kähler, V. A. Prisacariu, C. Y. Ren, X. Sun, P. Torr, and D. Murray, "Very high frame rate volumetric integration of depth images on mobile devices," *IEEE transactions on visualization and computer graphics*, vol. 21, no. 11, pp. 1241–1250, 2015.

[29] T. Whelan, S. Leutenegger, R. Salas-Moreno, B. Glocker, and A. Davison, "Elasticfusion: Dense slam without a pose graph." Robotics: Science and Systems, 2015.

[30] W. Wang, Y. Hu, and S. Scherer, "Tartanvo: A generalizable learning-based vo," in *Conference on Robot Learning*. PMLR, 2021, pp. 1761–1772.

[31] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *arXiv preprint arXiv:1804.02767*, 2018.