A Reliability-Aware Visual-Inertial Odometry for Dynamic and Low-Texture Environments

Yelu Liu[1], Ruokun Qu[2]*, Mengcheng Xu[3], Chenglong Li[4], Hui Jiang[5]
Engineering Technology Training Center, Civil Aviation Flight University of China, Chengdu 610000, China[1]
College of Air Traffic Management, Civil Aviation Flight University of China, Chengdu 610000, China[2,3]
School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu 610054, China[2,5]
College of Flight Technology, Civil Aviation Flight University of China, Guanghan 618307, China[4]

***Abstract*—Visual-inertial odometry (VIO) tends to degrade in aggressive dynamic and low-texture environments, where rapid motion, weak visual structure, and moving objects reduce the reliability of visual observations. This study presents TRAIL-VIO, a temporal reliability-aware visual-inertial odometry framework with line feature enhancement. The method estimates temporal observation reliability by combining semantic priors with IMU-based motion consistency, which allows a continuous and time-varying assessment of observation quality instead of frame-wise decisions. A reliability-aware point-line association scheme is also introduced, where inertial prediction is used to constrain feature matching and partially corrupted line segments are selectively retained. In addition, a reliability-guided marginalization strategy is applied to reduce the influence of unreliable visual constraints before they are incorporated into the prior. Experiments on the EuRoC MAV benchmark and a self-collected UAV dataset show that TRAIL-VIO achieves average RMSE values of 0.042 m and 9.51 m, respectively, outperforming representative baseline methods in dynamic and low-texture scenarios. Additional ablation, parameter-sensitivity, and runtime analysis further verify the contribution of the main modules, the robustness of the selected parameters, and the computational feasibility of the proposed framework.**

***Keywords*—Visual-inertial odometry; point-line feature fusion; dynamic environments; observation reliability estimation; marginalization; UAV navigation**

I. Introduction

Visual-inertial odometry (VIO) is widely used for autonomous navigation of micro aerial vehicles (UAVs), mobile robots, and other platforms in GPS-denied environments. By combining geometric constraints from visual measurements with short-term motion information from inertial sensors, VIO enables accurate and continuous state estimation [1], [2].

However, in real-world environments, VIO systems often operate under non-ideal conditions. Aerial platforms often undergo rapid motion, which easily introduces motion blur and large viewpoint changes. At the same time, many environments are weakly textured and contain moving objects. These factors not only degrade feature quality, but also make visual observations less reliable. As a result, incorrect data association may occur, leading to reduced estimation accuracy or even tracking failure. Improving robustness under such aggressive and dynamic conditions is still challenging.

Existing methods for improving VIO robustness mainly follow two directions. One line of work focuses on enhancing the estimator, typically through tightly coupled visual-inertial optimization. Another line introduces additional geometric features, such as line features, to complement point features [3]. These strategies do improve performance in difficult scenarios. However, they still depend heavily on the quality of front-end observations. When motion becomes aggressive, both point and line features can degrade before the back-end has a chance to compensate.

To address the limitations of the static-world assumption, recent studies have explored dynamic-scene SLAM and VIO methods based on semantic segmentation and geometric consistency checks [4], [5], [6]. These approaches aim to detect and filter out dynamic observations. Semantic cues help identify potentially movable objects, while geometric verification and IMU-based reasoning provide additional constraints. Together, they improve robustness to a certain extent.

While these dynamic-scene pipelines can remove obvious outliers, they still rely heavily on frame-wise filtering. This makes them sensitive to segmentation errors and short-term changes in object motion. The limitation becomes more evident in structurally rich environments. In low-texture scenes, line features often play an important role. Even partial occlusion or slight dynamic interference can cause an entire line segment to be discarded. As a result, useful structural information is unnecessarily lost [6]. Most existing methods handle uncertain observations only at the front end, typically through simple down-weighting. Their influence on the back-end is rarely considered. Without a temporal evaluation of observation reliability, these imperfect measurements can still enter the optimization process. Over time, they are absorbed into the marginalized prior, which gradually degrades the consistency of the sliding-window estimation [7].

These limitations suggest that the key issue is not only identifying dynamic observations, but also evaluating their reliability over time. In aggressive scenarios, semantic predictions, geometric residuals, and image quality can vary significantly across frames. Relying on frame-wise decisions is therefore insufficient. A more reasonable approach is to estimate observation reliability in a temporal and recursive manner, and incorporate it consistently into both front-end processing and back-end optimization.

To address these issues, we present TRAIL-VIO, a

*Corresponding author

reliability-aware framework tailored for aggressive dynamic environments. Instead of treating observation quality as a simple filtering problem at the front end, our method estimates it over time and uses it throughout the system. The estimated reliability is used in feature association, local structure handling, and marginalization. In this way, unreliable observations can be handled earlier, which helps maintain stable tracking while preserving long-term consistency. The main contributions of this study are summarized as follows:

- We introduce a recursive method to estimate temporal observation reliability. By combining semantic cues with IMU-based motion consistency, reliability is modeled as a time-evolving state rather than independent frame-wise decisions. This leads to more stable behavior under dynamic disturbances.
- We propose a reliability-aware front-end for point-line association. IMU prediction is used to guide matching, and partially corrupted line features are handled more carefully. Instead of discarding them completely, the method preserves valid segments when possible, which improves robustness in low-texture scenes.
- We develop a marginalization strategy that takes observation reliability into account. Observations are screened before being incorporated into the prior, reducing the influence of degraded measurements. This helps limit error accumulation and improves long-term estimation consistency.

The rest of this study is structured as follows: Section II reviews the related work. Section III outlines the system architecture. Section IV gives the details of our proposal. Section V details the experimental results, and Section VI concludes the study and discusses potential directions for future work.

II. Related Work

A. Visual-Inertial Estimation and Feature Representations

Visual-inertial odometry typically advances on two fronts, the underlying state estimator and the choice of visual features. Point-based systems remain the standard. Frameworks like VINS-Mono [1] and ORB-SLAM3 [2] provide reliable starting points for most state estimation tasks. Yet, points alone often struggle in environments lacking distinct textures. To address this, systems such as PL-VIO [3] bring in lines or other geometric primitives, showing noticeable gains in structurally challenging scenes.

Early VIO architectures relied heavily on filtering techniques. MSCKF [8] set a benchmark here by introducing a multi-state constraint Kalman filter for vision-aided navigation. Later variants like S-MSCKF [9] pushed this concept further to better handle aggressive motion. Alternatively, ROVIO [10] tied direct image alignment into an EKF framework, which helps when standard feature tracking fails. These filter-based methods are fast. They fit well on resource-constrained hardware. Still, their accuracy tends to drop if visual observations degrade even for a short period.

Over time, the field shifted heavily toward nonlinear optimization. Keyframe-based, tightly coupled optimization proved highly effective in systems like OKVIS [11]. More recently, frameworks like OpenVINS [12] and VINS-Fusion [13] have generalized these concepts, making sliding-window optimization a common setup for multi-sensor fusion. In mostly static or visually clear environments, these methods are highly accurate. The catch lies in the front end. They generally trust that incoming visual constraints are reliable. When a platform moves aggressively through degraded conditions, this assumption often breaks down.

Beyond tweaking the backend estimator, researchers look to richer feature representations to bridge these reliability gaps. Point-line fusions are particularly common. PL-SLAM [14] and PL-VINS [15] early on showed how lines naturally complement points where textures are sparse. This sparked a wave of extensions. Works like Trifo-VIO [16], PLI-VINS [17], LRPL-VIO [18], PLE-SLAM [19] and PL-CVIO [20] adapt point-line constraints to various sensor setups and scenarios. Others look even higher up the geometric hierarchy. StructVIO [21] leverages structural regularities common in man-made spaces, while Pop-up SLAM [22] incorporates planar structures. Ultimately, adding these higher-level primitives helps stabilize the system when simple point features fail.

While incorporating more geometric features certainly improves observability, it does not automatically guarantee reliability. Real-world conditions, such as sudden lighting shifts, partial occlusions, or motion blur, can easily corrupt both point and line tracking. Lines present a unique challenge here. Most existing frameworks treat them as rigid, all-or-nothing entities. A line segment is typically either fully accepted into the optimization or discarded entirely, even if only a small section is corrupted. This binary approach often wastes valuable geometric data in cluttered environments.

Simply having more constraints available is not enough. A robust system must also be able to gauge which constraints to actually trust over time. This becomes particularly critical in dynamic scenarios. In such environments, the observation quality of a tracked feature can fluctuate wildly from one frame to the next. If the front end fails to filter out these unreliable observations, the errors inevitably bleed into the backend. Once they contaminate the state estimation and the marginalization process, the long-term accuracy of the system gradually degrades.

B. VIO/SLAM in Dynamic Environments

Navigating dynamic environments remains a central challenge for VIO and SLAM systems. Initial efforts leaned heavily on pure vision. Systems like DS-SLAM [23], DynaSLAM [4], DynaSLAM II [24], and SOF-SLAM [25] use semantic segmentation and motion consistency to detect and mask dynamic regions before they can corrupt pose estimation. Taking this a step further, object-level methods such as visual-inertial multi-instance SLAM [26] attempt to jointly model camera motion alongside moving objects and scene structure. Clearly, blending semantic and geometric cues helps stabilize localization when the environment is moving.

However, a key issue lies in how dynamic observations are handled. Most frameworks rely on frame-by-frame judgments. They classify a visual observation as either valid or invalid based on an instantaneous snapshot—such as a single

segmentation mask or immediate motion inconsistency. This works well when dynamic evidence is obvious. Yet, it quickly becomes brittle. Segmentation masks can be noisy. Motion is sometimes intermittent, and dynamic objects often stop moving temporarily. In aerial scenarios, these problems are amplified. Rapid viewpoint shifts and motion blur cause the quality of observations to swing unpredictably between consecutive frames.

To ground these fluctuating visual signals, researchers increasingly turn to inertial priors. By explicitly fusing IMU data with semantic and geometric cues, systems like DynaVINS [5] and D-VINS [6] build stronger defenses against dynamic interference. Extensions of this idea tackle specific edge cases: VINS-Dimc [27] applies multiple constraints, SRVIO [28] targets loop closures in moving scenes, and RD-VIO [29] uses IMU-informed matching to verify motion types. GMS-VINS [30] also shows the value of adapting semantic processing on the fly. Collectively, this body of work confirms that inertial data is essential for cross-checking visual dynamics.

But the front end is only half the battle. Long-term accuracy hinges heavily on back-end consistency. DM-VIO [7] illustrates this perfectly: how a system handles marginalization directly dictates its resilience over time. If the front end fails to adequately suppress a moving feature, that corrupted observation sneaks into the sliding-window optimization. Once it gets baked into the historical prior, the error becomes permanent, slowly degrading the entire trajectory.

Several limitations remain in current approaches. First, we still rely too much on short-sighted, frame-level suppression, leaving estimators vulnerable to transient errors like segmentation flickering. Furthermore, observation reliability is rarely tracked over time. This is especially true for heterogeneous data like point-line combinations. As noted earlier, partially corrupted lines are often handled clumsily—either kept entirely or thrown away, sacrificing useful geometric fragments. Finally, there is a systemic failure to protect the marginalized prior. While many algorithms down-weight bad features in the current optimization window, few provide a structural mechanism to stop these unreliable observations from polluting historical estimates.

These observations suggest that a more unified strategy is needed. To truly handle aggressive dynamic environments, the system needs a temporally consistent way to measure observation reliability. More importantly, this reliability metric must be woven seamlessly through both front-end feature association and back-end marginalization.

III. System Overview

The proposed framework focuses on improving the robustness of visual-inertial estimation in dynamic and low-texture environments, where unreliable visual observations may pass through the front end and affect back-end optimization. We break away from the standard practice of frame-wise binary feature rejection. Instead, we introduce a unified reliability model. This model evaluates the quality of each point and line over time, and the resulting reliability is used throughout the estimation pipeline. Fig. 1 outlines the overall architecture.

The pipeline ingests three primary streams of data, which are raw images, high-frequency IMU measurements, and semantic segmentation masks. Each plays a distinct role. Images yield the base point and line features. The IMU provides short-term kinematic predictions, serving as a physical baseline for geometric consistency checks. Meanwhile, the semantic masks supply category-level priors to flag potentially moving objects. Rather than processing these inputs in isolation, the system fuses them to dynamically estimate a continuous reliability score for each visual feature.

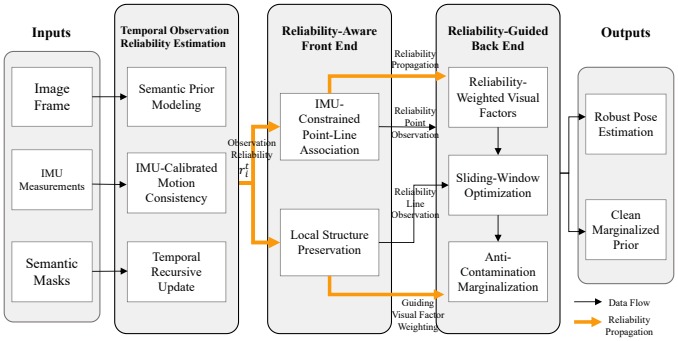


Fig. 1. Overall framework of the proposed method. Temporal observation reliability is estimated using semantic and inertial cues, and is further utilized in both front-end association and back-end marginalization.

We build upon a standard sliding-window visual-inertial formulation, tracking the vehicle's orientation, position, velocity, and sensor biases at each keyframe. Where our approach diverges is in the front-end processing. As points and lines are extracted, they are not just blindly tracked. They are assigned a temporally updated reliability score. We compute this by cross-referencing semantic priors with IMU-driven geometric verification, smoothing the results recursively over time.

This metric directly steers the front-end data association. Highly reliable features are actively preserved to anchor the local structure. Conversely, suspicious observations are suppressed before they can trigger faulty matches. Notably, this continuous scoring allows the system to salvage partially corrupted line features, preserving their valid segments rather than simply discarding the entire line.

The reliability-aware philosophy extends naturally into the back end. During optimization, the same confidence scores modulate the visual factors. Trustworthy features dominate the objective function. Uncertain ones are heavily down-weighted, or outright pruned, long before they can be absorbed into the marginalized prior. By stopping transient errors from accumulating within the sliding window, the system maintains stable, long-term consistency, even during aggressive maneuvers typical in UAV flight.

IV. Proposed Method

A. Temporal Observation Reliability Estimation for Aggressive Dynamic Scenarios

In aggressive UAV scenarios, both semantic predictions and geometric observations tend to become unstable. Semantic segmentation may produce incorrect labels or low-confidence outputs. At the same time, geometric measurements are easily affected by motion blur, rapid rotations, and depth ambiguity. These factors often appear together. As a result, decisions

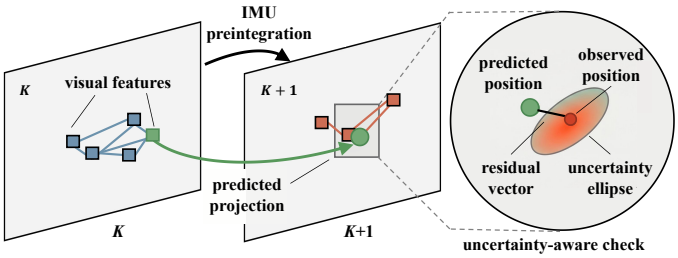


Fig. 2. IMU-calibrated kinematic consistency. A feature from the previous frame is projected to the current frame using IMU-preintegrated motion, and the difference between predicted and observed positions is evaluated with uncertainty.

made independently at each frame can be unreliable. Rather than forcing observations into binary categories, we treat the problem from a temporal perspective. Each observation is assigned with a continuous reliability value, which evolves over time. This value is not determined by a single cue. Instead, it is gradually refined by combining semantic priors, inertial consistency, and temporal recursion. This formulation can make the estimation less sensitive to short-term disturbances.

1) Semantic prior modeling: To introduce semantic cues without adding significant overhead, a lightweight YOLOv8n-seg model [31], pretrained on the MS COCO dataset [32], is employed. The semantic output is not used for direct filtering. Instead, it provides an initial estimate of observation reliability.

For each observation $\mathbf{u}_i^t$, the network outputs a category label c_i^t and a confidence score q_i^t. Each category is associated with a predefined coefficient $\rho_c \in [0,1]$, reflecting how likely it is to remain static. Static structures such as buildings are assigned higher values, while potentially dynamic objects are assigned lower ones. The semantic prior is then computed as Eq. (1):

$$s_i^t = \rho_{c_i^t} q_i^t + (1 - q_i^t) s_0 \tag{1}$$

where, s_0 is a neutral prior, typically set to 0.5. When the confidence is high, the model relies more on the semantic prediction. When it is low, the estimate falls back toward a neutral value. Compared with binary masking, this formulation keeps uncertainty in the representation. It also provides a smoother starting point for later updates.

2) IMU-calibrated kinematic consistency: Semantic information alone is often insufficient to determine whether an observation is truly static. To complement it, an IMU-based consistency check is introduced. The core idea is to evaluate whether a visual observation follows the rigid-body motion predicted by inertial measurements.

As illustrated in Fig. 2, IMU measurements between two consecutive frames are first preintegrated to estimate relative motion. A feature observed in the previous frame is then projected into the current frame [see Eq. (2)]:

$$\hat{\mathbf{u}}_i^t = \pi\left(\mathbf{T}_{cb}\hat{\mathbf{T}}_{t\leftarrow t-1}^{\text{imu}}\mathbf{T}_{bc}\mathbf{P}_i^{t-1}\right) \tag{2}$$

The difference between the predicted and observed positions forms the reprojection residual. Instead of applying a fixed threshold, this residual is evaluated with respect to its uncertainty. IMU noise, depth uncertainty, and image noise are all taken into account. The consistency is measured using the Mahalanobis distance [see Eq. (3)]:

$$m_i^t = (\mathbf{e}_i^t)^\top (\mathbf{\Sigma}_i^t)^{-1} \mathbf{e}_i^t \tag{3}$$

This allows the evaluation to adapt to different motion conditions. When motion is aggressive, uncertainty increases and the constraint becomes more tolerant. Under stable conditions, the check becomes stricter. In this way, the measure remains consistent with the underlying physical model.

3) Temporal recursion and reliability update: To reduce the impact of transient errors and occasional inconsistencies, reliability is updated over time rather than determined from a single frame. Each observation is treated as a temporal state. Three conditions are considered static, quasi-static, and dynamic. At each time step, the predicted state from the previous frame is combined with current evidence. This is done through a Bayesian-style update [see Eq. (4)]:

$$\mathbf{p}_i^t = \frac{\mathbf{l}_i^t \odot \bar{\mathbf{p}}_i^t}{\mathbf{1}^\top (\mathbf{l}_i^t \odot \bar{\mathbf{p}}_i^t)} \tag{4}$$

The update blends historical information with new observations. Short-term noise is less likely to dominate the result. The estimation therefore becomes more stable over time. A scalar reliability value is derived as Eq. (5):

$$r_i^t = P(z_i^t = S) + \eta P(z_i^t = Q) \tag{5}$$

where, $\eta \in [0,1]$ controls how much partially reliable observations contribute. The estimated reliability is used throughout the system. In the front end, it influences feature selection and association. In the back end, it is incorporated into visual factors and marginalization.

B. Reliability-Aware Point-Line Association and Local Structure Preservation

This module transforms the estimated observation reliability into robust front-end constraints. In aggressive UAV scenarios, rapid motion, viewpoint changes, and weak texture often reduce the stability of point features. Line features can provide complementary structural information, but they are sensitive to partial occlusion and local dynamic interference. To address these challenges, the proposed method integrates IMU-guided search, reliability-aware association, and local structure preservation to retain valid observations while suppressing unreliable ones.

1) IMU-constrained adaptive matching for point and line features: The front-end operates feature extraction based on ORB points and LSD line. For point features, the expected position in the current frame is predicted using IMU-preintegrated motion. Matching is then performed within a local search region centered on this prediction, rather than globally. The size of this region is adjusted based on motion uncertainty. Under stable motion conditions, the region is kept small to maintain efficiency and matching precision.

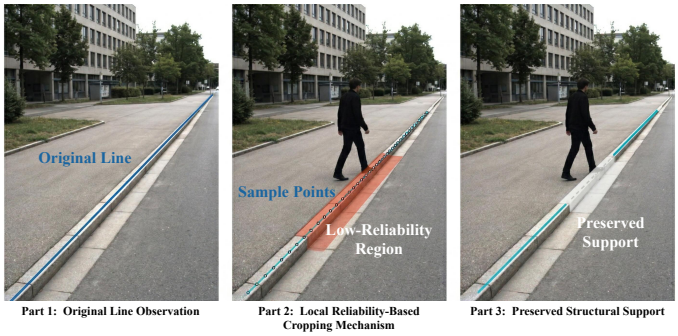


Fig. 3. Local line cropping and structure preservation mechanism. The reliability map is sampled along each matched line segment, and only the connected subregion with sufficiently high reliability is retained as the valid structural support.

During aggressive motion, the region is expanded to prevent the loss of valid correspondences. Line features are handled with a similar predictive approach. Each line segment from the previous frame is projected into the current frame, and candidate matches are searched within an uncertainty-aware band around the predicted location. This limits mismatches caused by viewpoint changes and assists tracking in low-texture scenes.

Observation reliability is also incorporated into the matching cost. For line features, the cost function is defined as Eq. (6):

$$C_k = \lambda_d d_{\mathrm{LBD}} + \lambda_\theta d_\theta + \lambda_\perp d_\perp + \lambda_r (1 - r^t_{\ell_c}) \tag{6}$$

The first three terms represent descriptor similarity and geometric consistency, while the final term applies a penalty based on reliability. As a result, matches exhibiting both geometric consistency and temporal reliability are prioritized. A similar reliability-based filtering mechanism is applied to point features to select stable candidates.

2) Local line cropping and structure preservation: Most existing methods treat line segments as complete entities, discarding the entire line if a portion overlaps with a dynamic region. This strategy can be overly restrictive in dynamic environments where lines may only be partially affected. The proposed local line cropping mechanism is illustrated in Fig. 3.

Instead of binary rejection, the proposed method evaluates lines locally. A reliability map is constructed on the image plane, and reliability values are sampled along the line segment. The average reliability is computed as Eq. (7):

$$r^t_\ell = \frac{1}{N} \sum_{s=1}^{N} r(\mathbf{x}_s) \tag{7}$$

While this average provides an indication of overall quality, it is not used directly for filtering. The line is parameterized, and sub-regions are evaluated individually. Only the segments with reliability exceeding a defined threshold are retained [see Eq. (8)]:

$$\Omega_{\mathrm{stat}} = \{ s \in [0,1] \mid r(\ell(s)) > \tau_{\mathrm{keep}} \} \tag{8}$$

Among the retained sections, the longest continuous segment is selected as the valid observation, and the remainder is removed. This makes it possible to preserve structural information even if the original line is partially degraded.

3) Joint reliability-aware optimization: After data association, observation reliability is further integrated into the optimization stage. The contribution of each observation is weighted by both its geometric residual and its estimated reliability. For point features, the reprojection residual is defined as Eq. (9):

$$\mathbf{e}_{p,i} = \mathbf{u}_i - \hat{\mathbf{u}}_i \tag{9}$$

For line features, the residual is computed using the preserved cropped segments, limiting the influence of unreliable portions. The joint optimization problem is formulated as Eq. (10):

$$\min_{\mathcal{X}} \sum_{i \in \mathcal{P}} w_{p,i} \rho \left(\|\mathbf{e}_{p,i}\|^2_{\mathbf{\Sigma}^{-1}_{p,i}} \right) + \sum_{k \in \mathcal{L}} w_{l,k} \rho \left(\|\mathbf{e}_{l,k}\|^2_{\mathbf{\Sigma}^{-1}_{l,k}} \right) \tag{10}$$

The $w_{p,i}$ and $w_{l,k}$ weights are determined by the temporal reliability. Consistent observations receive higher weights, while less reliable measurements are down-weighted.Robust kernels ρ are maintained to filter instantaneous outliers. The reliability weighting complements the robust kernel by reflecting the historical stability of the observation, which adds another way to handle dynamic interference during sliding-window estimation.

C. Reliability-Guided Anti-Contamination Marginalization

In sliding-window visual-inertial odometry, marginalization is used to keep the state size bounded and maintain computational efficiency. In dynamic or aggressive scenarios, however, some unreliable visual observations may pass front-end checks and enter the marginalized prior. Once included, these corrupted constraints can accumulate and lead to estimation drift over time. A reliability-guided marginalization strategy is introduced to reduce this effect. Observation reliability estimated in the front end is propagated to the prior construction stage, where visual factors with low reliability are down-weighted before marginalization. The overall pipeline is shown in Fig. 4.

Let $r_j \in [0,1]$ denote the reliability of the j-th visual factor. For point features, this value comes from temporal reliability estimation. For line features, it is derived from the retained line segment. Rather than using a hard threshold, a continuous gating function is applied to control the contribution of each factor [see Eq. (11)]:

$$\Gamma(r_j) = \begin{cases} 0, & r_j \le \tau_{\mathrm{drop}}, \\ \dfrac{1}{1 + \exp\left[-k\left(r_j - \tau_{\mathrm{marg}}\right)\right]}, & \tau_{\mathrm{drop}} < r_j < \tau_{\mathrm{safe}}, \\ 1, & r_j \ge \tau_{\mathrm{safe}}. \end{cases} \tag{11}$$

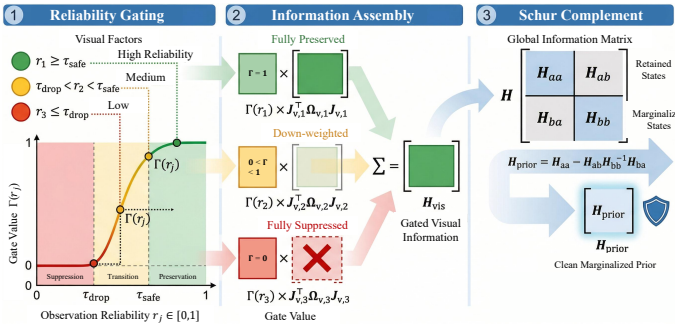


Fig. 4. Reliability-guided anti-contamination marginalization strategy. Observation reliability estimated in the front end is propagated to prior construction, so that suspicious visual constraints are weakened or excluded before being absorbed into the marginalized prior.

This function separates factors into three cases, low-reliability factors are removed, high-reliability ones are kept, and intermediate cases are smoothly weighted. Compared with binary selection, this formulation reduces sensitivity near the threshold. The gating is applied during linearization by scaling each visual factor. The visual information matrix becomes Eq. (12):

$$\mathbf{H}_{\text{vis}} = \sum_{j} \Gamma(r_j)\, \mathbf{J}_{v,j}^{\top} \mathbf{\Omega}_{v,j} \mathbf{J}_{v,j} \tag{12}$$

In this way, reliability is incorporated directly into the information matrix. Visual factors with low reliability contribute less to the optimization. After assembling the gated system, marginalization is carried out using the standard Schur complement. The state increment is partitioned into marginalized variables $\delta \mathbf{X}b$ and retained variables $\delta \mathbf{X}a$, yielding the prior [see Eq. (13)]:

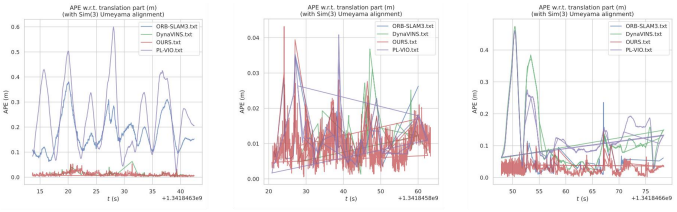
$$\mathbf{H}_{\text{prior}} = \mathbf{H}_{aa} - \mathbf{H}_{ab}\mathbf{H}_{bb}^{-1}\mathbf{H}_{ba} \tag{13}$$

Since unreliable factors are suppressed beforehand, the resulting prior mainly reflects reliable visual constraints and inertial information. A small damping term can be added if needed to improve numerical stability. Compared with conventional approaches, this method introduces a reliability-based filtering step before marginalization. This helps limit the influence of unreliable observations on the prior and improves consistency in dynamic environments.

V. Experimental Results

To evaluate the proposed method under different motion conditions and scene complexities, experiments are conducted on the EuRoC MAV benchmark and a self-collected UAV dataset. Both datasets provide synchronized image and IMU measurements, allowing the proposed framework to be evaluated as a complete visual-inertial odometry system. The evaluation focuses on scenarios involving rapid motion, weak texture, and dynamic interference.

The method is compared with several representative approaches, including ORB-SLAM3, DynaVINS, and PL-VIO.



(a) MH03 (b) V102 (c) V203

Fig. 5. APE of ORB-SLAM3, DynaVINS, PL-VIO, and the proposed method on the EuRoC dataset.

TABLE I. Comparison of the Average RMSE [m] of the Proposed Method with ORB-SLAM3, DynaVINS, and PL-VIO on the EuRoC Dataset.

Sequence	ORB-SLAM3	DynaVINS	PL-VIO	Ours
MH01	0.075	0.084	0.210	0.062
MH02	0.044	0.105	0.232	0.037
MH03	0.217	0.054	0.312	0.046
MH04	0.132	0.122	0.139	0.075
MH05	0.121	0.147	0.257	0.057
V101	0.059	0.047	0.059	0.049
V102	0.023	0.018	0.016	0.011
V103	0.096	0.180	0.130	0.037
V201	0.040	0.056	0.032	0.042
V202	0.062	0.090	0.095	0.021
V203	0.104	0.154	0.201	0.027
Average	0.088	0.096	0.153	0.042

The evaluation focuses on scenarios involving dynamic interference, weak texture, and large motion. For quantitative analysis, Absolute Pose Error (APE) and Absolute Trajectory Error (ATE) are used as the primary metrics. APE reflects the overall trajectory deviation, while ATE focuses on translational accuracy. RMSE is also reported to give a more compact summary and to capture sensitivity to larger errors.

A. Evaluation on the EuRoC Dataset

The EuRoC MAV dataset [33] is widely used in visual-inertial odometry research. It provides synchronized stereo images and IMU measurements, together with ground-truth trajectories. The dataset includes two indoor environments, Machine Hall (MH) and Vicon Room (V), covering motion patterns from relatively stable to aggressive.

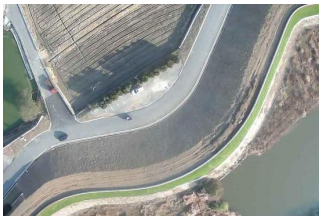
All 11 sequences are included in the evaluation, namely MH01-MH05, V101-V103, and V201-V203. These sequences involve various challenges, such as rapid motion, motion blur, repeated structures, and weak-texture regions. Some sequences are particularly demanding, which makes them suitable for testing robustness.

As can be seen in Fig. 5, the proposed method generally produces lower errors with reduced fluctuations compared to the baseline methods. On relatively stable sequences such as V102, the performance of all methods is quite close. On more challenging sequences like MH03 and V203, fewer error spikes can be observed, and the estimates remain more stable overall. This behavior is likely related to the temporal reliability mechanism, which helps mitigate the impact of degraded observations.

From Table I, the proposed method achieves an average RMSE of 0.042 m, which is lower than ORB-SLAM3, Dy-

(a) TR9S Hexacopter UAV

(b) TR9S lands around the lake



(c) Urban road data scene

(d) Forest and field data scene

Fig. 6. The data collection system and typical test scenes.

naVINS, and PL-VIO. The improvement becomes more noticeable on challenging sequences such as MH03, MH05, V103, V202, and V203, where motion complexity and dynamic interference are more pronounced. On sequences such as V101 and V201, the proposed method does not obtain the lowest error, although the differences remain relatively small. Taken together, the results suggest that the method maintains stable performance across different sequences and shows improved robustness under varying visual-inertial conditions.

B. Evaluation on Self-Collected UAV Flight Dataset

To further examine the method under real flight conditions, we collected a UAV dataset using a hexarotor platform equipped with a monocular camera, an IMU, and a differential GPS/RTK system. The camera and IMU data are used for odometry estimation, while the RTK measurements provide ground truth. The camera runs at 30Hz and the IMU at 100Hz. All sensors are time-synchronized, and the camera–IMU extrinsics are calibrated offline. For consistency, all baseline methods use the same input data, and trajectory errors are computed against the RTK reference.

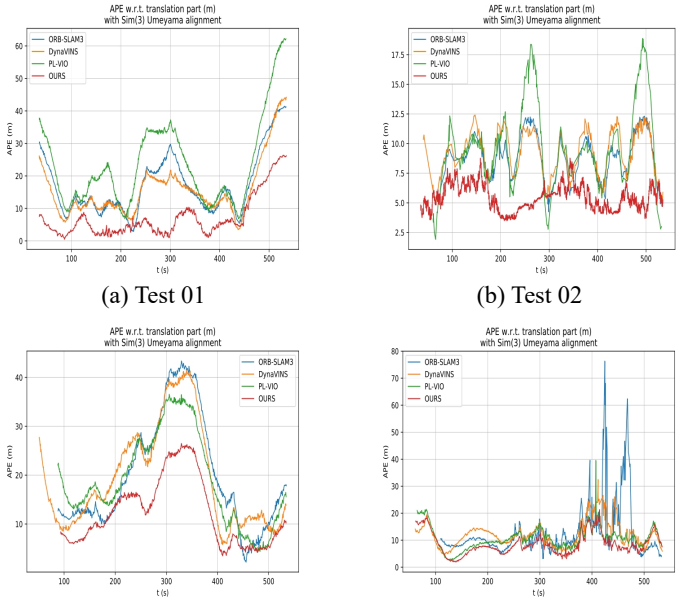
Five sequences, denoted as Test01-Test05, are recorded in different environments, including urban areas, lakes, forests, and suburban roads. The total flight distance is about 20km. These sequences include a range of conditions such as aggressive motion, weak-texture regions, and dynamic objects like vehicles and pedestrians. The data collection platform and representative test scenes are shown in Fig. 6.

Trajectory evaluation is carried out using the EVO toolkit after aligning the estimated trajectories with the RTK ground truth.

From Table II, the proposed method achieves an average RMSE of 9.51 m, which is lower than ORB-SLAM3, DynaVINS, and PL-VIO. The difference is more visible on sequences such as Test01, Test04, and Test05, where longer trajectories, stronger motion, and dynamic interference make estimation more difficult.

TABLE II. COMPARISON OF RMSE [m] OF THE PROPOSED METHOD WITH ORB-SLAM3, DYNAVINS, AND PL-VIO ON THE UAV DATASET.

Sequence	ORB-SLAM3	DynaVINS	PL-VIO	Ours
Test01	30.52	17.79	27.41	8.45
Test02	8.75	9.65	10.80	5.42
Test03	10.56	10.95	42.68	9.87
Test04	36.09	23.25	22.29	14.55
Test05	46.14	12.79	11.46	9.26
Average	26.41	14.89	22.93	9.51



(a) Test 01 (b) Test 02

(c) Test 04 (d) Test 05

Fig. 7. APE curves of ORB-SLAM3, DynaVINS, PL-VIO, and the proposed method on the self-collected UAV dataset.

As seen in Fig. 7, the proposed method generally maintains lower errors and smoother trends along the trajectories. The baseline methods show larger fluctuations in several segments, together with occasional spikes, which suggests local tracking instability.

This difference becomes clearer in segments with rapid motion or degraded visual input. In such cases, ORB-SLAM3 and DynaVINS tend to produce noticeable error increases and drift over time, while the proposed method remains more stable and avoids abrupt degradation. This behavior indicates that the reliability-aware design can help improve consistency over longer trajectories in dynamic outdoor scenes.

C. Ablation Study

To further analyze the contribution of the proposed components, two ablation studies are conducted on the self-collected UAV dataset. The reported RMSE values are averaged over the five UAV sequences. These sequences include outdoor scenes with weak textures, camera motion, and dynamic interference, and therefore provide a practical basis for evaluating the stability of each module.

Table III presents the ablation results of the reliability-aware VIO framework on the challenging UAV sequences.

TABLE III. ABLATION STUDY OF THE PROPOSED RELIABILITY-AWARE VIO FRAMEWORK.

Temporal Observation Reliability Estimation	Point-Line Association and Local Structure Preservation	Reliability-Guided Marginalization	RMSE (m)	Relative percentage Reduction in RMSE
-	-	-	22.93	-
✓	-	-	15.09	34.19%
-	✓	-	17.64	23.07%
-	-	✓	13.33	41.87%
✓	✓	✓	9.51	58.53%

TABLE IV. ABLATION STUDY OF THE TEMPORAL RELIABILITY ESTIMATION MODULE.

Semantic Prior	IMU-Based Motion Consistency	RMSE (m)	Relative percentage reduction in RMSE
-	-	20.39	-
✓	-	16.58	18.69%
-	✓	15.24	25.26%
✓	✓	10.75	47.28%

Each module brings a certain improvement over the base point-line VIO system. Temporal observation reliability estimation reduces the average RMSE from 22.93 m to 15.09 m, which suggests that modeling the credibility of visual observations over time helps suppress unstable measurements. The point-line association and local structure preservation strategy also improves the result by keeping useful geometric information from partially affected line features. Reliability-guided marginalization gives the largest individual reduction among the three components. Preventing unreliable visual constraints from being absorbed into the marginalized prior is helpful for limiting accumulated drift. With all three components enabled, TRAIL-VIO achieves the lowest average RMSE of 9.51 m.

Table IV reports the average performance of the temporal reliability estimation module across the five UAV sequences. Using semantic priors alone improves the RMSE from 20.39 m to 16.58 m, mainly because potentially dynamic regions can be identified before they strongly affect pose estimation. IMU-based motion consistency alone leads to a slightly lower RMSE of 15.24 m, showing that inertial prediction provides a useful motion-level check when visual observations become uncertain. The combination of both cues achieves the best average result. Semantic information and IMU-based consistency are not simply redundant, and they tend to compensate for each other in temporal reliability estimation.

D. Parameter Sensitivity Analysis

To address potential concerns regarding the generalizability of the manually configured parameters and to evaluate the robustness of the proposed framework, a comprehensive sensitivity analysis is conducted. We select two highly representative and challenging sequences for this evaluation: MH05 from the EuRoC dataset, which features indoor aggressive motion and low-illumination, and Test04 from the self-collected UAV dataset, which encompasses long-distance outdoor flight with strong dynamic interference. The APE RMSE is employed as the primary evaluation metric.

1) Sensitivity to Reliability Gating Thresholds: The gating parameters τ_{drop} and τ_{safe} control the reliability-guided marginalization strategy in Eq. (11), where visual factors are divided into suppression, transition, and preservation regions according to their estimated reliability. To examine their individual effects, a one-factor-at-a-time control-variable strategy is adopted. In each test, only one threshold is varied, while the remaining parameters are kept fixed at their default values.

TABLE V. SENSITIVITY OF APE RMSE [m] TO THE DROP THRESHOLD τ_{drop}.

τ_{drop}	MH05	Test04
0.00	0.082	18.20
0.05	0.071	16.45
0.10	0.063	15.10
0.15	0.058	14.62
0.20	0.057	14.55
0.25	0.057	14.56
0.30	0.058	14.58
0.35	0.059	14.65
0.40	0.064	15.30
0.45	0.076	17.85
0.50	0.095	21.40

TABLE VI. SENSITIVITY OF APE RMSE [m] TO THE SAFE THRESHOLD τ_{safe}.

τ_{safe}	MH05	Test04
0.50	0.065	15.80
0.55	0.062	15.30
0.60	0.060	14.95
0.65	0.059	14.70
0.70	0.058	14.58
0.75	0.057	14.56
0.80	0.057	14.55
0.85	0.057	14.55
0.90	0.058	14.57
0.95	0.059	14.62
1.00	0.063	15.15

First, the influence of the lower suppression threshold τ_{drop} is evaluated. τ_{safe} and the steepness factor k are fixed, and τ_{drop} is varied from 0.0 to 0.5. The results are reported in Table V. When τ_{drop} is close to 0.0, the suppression effect becomes weak. As a result, some unreliable observations caused by motion blur, dynamic objects, or inconsistent feature associations may still contribute to the optimization and the marginalized prior, leading to an increase in APE RMSE. In contrast, when τ_{drop} is set too high, especially above 0.45, the system tends to suppress more visual factors than necessary. This may remove valid geometric constraints and weaken the visual support for pose estimation. A relatively stable region can be observed when τ_{drop} lies in the interval $[0.15, 0.35]$. Within this range, the RMSE values remain close to the baseline results, approximately $0.057\,\mathrm{m}$ on MH05 and $14.55\,\mathrm{m}$ on Test04.

A similar analysis is conducted for the upper preservation threshold τ_{safe}. In this experiment, τ_{drop} is fixed at its default value, and τ_{safe} is swept from 0.5 to 1.0. As shown in Table VI, stable performance can be observed within a relatively broad interval, and the performance does not collapse under moderate parameter perturbations. Therefore, TRAIL-VIO does not depend on a narrowly selected pair of gating parameters. The default values of τ_{drop} and τ_{safe} are located within stable performance regions rather than at isolated optimal points.

2) Sensitivity to the Transition Steepness Factor: After analyzing the two reliability gating thresholds, we further evaluate the effect of the transition steepness factor k. The steepness factor k controls the shape of the sigmoid weighting function in Eq. (11). It determines how rapidly the reliability

TABLE VII. SENSITIVITY OF APE RMSE [m] TO THE TRANSITION STEEPNESS FACTOR k.

k	MH05	Test04
1	0.063	15.12
5	0.058	14.65
10	0.057	14.55
15	0.057	14.59
20	0.059	14.72
50	0.065	15.38

weight changes within the transition region between τ_{drop} and τ_{safe}. A small k produces a smoother transition, while a large k makes the weighting function closer to a hard switch. Therefore, this parameter affects how gently or aggressively visual factors are down-weighted when their reliability values fall between the suppression and preservation thresholds. In this experiment, τ_{drop} and τ_{safe} are fixed at their default values, and only k is varied. The tested values are selected as $k = \{1, 5, 10, 15, 20, 50\}$, covering a wide range from nearly linear transition to a steep, almost binary gating behavior.

The results in Table VII show that a moderate steepness factor generally leads to more stable performance. Setting k to an extremely low value like 1 tends to under-penalize questionable observations. The transition becomes too flat. Consequently, some degraded features retain a disproportionate amount of influence during the information matrix assembly, which causes a slight increase in the tracking error. And pushing k to 50 mimics a rigid binary threshold. This sharp cutoff can abruptly discard features that are only experiencing transient ambiguity. Such sudden losses of structural constraints occasionally introduce minor instabilities into the sliding-window optimization.

The proposed framework demonstrates considerable tolerance between these two extremes. For values of k roughly between 5 and 15, the APE RMSE remains largely consistent. On the MH05 sequence, the error fluctuates by merely a few millimeters within this moderate range. The Test04 sequence shows a similarly stable plateau. This behavior implies that the soft-gating mechanism successfully bridges the gap between total suppression and full preservation. By smoothing the factor weighting, the system generally avoids overreacting to short-term observation noise.

3) Tolerance to Semantic Prior Perturbation: In addition to the reliability gating parameters, we further evaluate the influence of the semantic category coefficient ρ_c. In the proposed framework, ρ_c represents the prior static reliability of a semantic category in the semantic reliability model. Static structures, such as buildings, roads, and background regions, are assigned relatively high prior values, while potentially dynamic objects, such as pedestrians and vehicles, are assigned lower prior values. Since these coefficients are manually predefined, it is necessary to examine whether inaccurate semantic priors would cause a noticeable degradation in trajectory estimation.

To simulate possible miscalibration of semantic priors in unseen environments, a perturbation factor Δ_ρ is introduced to the default semantic coefficient. The perturbed coefficient is defined as Eq. (14):

$$\tilde{\rho}_c = \text{clip}(\rho_c + \Delta_\rho, 0, 1), \tag{14}$$

TABLE VIII. SENSITIVITY OF APE RMSE [m] TO SEMANTIC PRIOR PERTURBATION.

Perturbation Δ_ρ	MH05	Test04
-30%	0.066	15.32
-20%	0.062	14.95
-10%	0.059	14.68
0	0.057	14.55
$+10\%$	0.058	14.65
$+20\%$	0.061	14.88
$+30\%$	0.065	15.25

where, $\tilde{\rho}_c$ denotes the perturbed semantic coefficient, and $\text{clip}(\cdot)$ limits the perturbed value to the valid interval $[0, 1]$. In this experiment, Δ_ρ is swept from -0.30 to $+0.30$ with a step size of 0.10. The case of $\Delta_\rho = 0$ represents the default semantic prior setting used in the main experiments.

During this analysis, all semantic categories are divided into two broad groups, static categories and potentially dynamic categories. The same perturbation strategy is applied to the semantic coefficients of these categories to evaluate the tolerance of the system to prior uncertainty. This setting is intentionally simple. It does not aim to find category-specific optimal weights. Instead, it tests whether the proposed reliability estimation framework remains stable when the manually assigned semantic priors are not fully accurate. The other parameters, including τ_{drop}, τ_{safe}, and k, are fixed at their default values.

The results are reported in Table VIII. When negative perturbations are introduced, the semantic reliability assigned to static categories decreases, and some stable background features may contribute less to the visual constraints. Positive perturbations have the opposite effect. They may increase the reliability of potentially dynamic or ambiguous regions, so these observations are less strongly penalized by the semantic prior. Both cases can affect the APE RMSE to some extent. Within the tested perturbation range, the variation remains limited, as the final observation reliability is still refined by IMU-based motion consistency and temporal reliability updating.

TRAIL-VIO is not overly sensitive to precisely tuned semantic category coefficients. Semantic information provides an initial prior, rather than a final reliability decision. The subsequent IMU-calibrated kinematic consistency and temporal update help reduce the influence of imperfect semantic priors. Therefore, the proposed framework shows a certain tolerance to manually assigned category weights and does not require highly precise semantic coefficient tuning for every new environment.

E. Runtime Analysis

To evaluate the computational overhead of TRAIL-VIO, the average runtime of the main modules is measured on a laptop equipped with an Intel Core i9-14900HX CPU, 31.1 GB RAM, and an NVIDIA GeForce RTX 4060 GPU under Ubuntu 18.04.6 LTS. The runtime is reported in milliseconds per frame and averaged on the self-collected UAV sequences.

We separate YOLOv8n-seg inference from temporal observation reliability estimation. It only includes the reliability assignment based on semantic outputs, IMU-based motion consistency evaluation, and temporal reliability updating.

TABLE IX. AVERAGE COMPUTATION TIME [ms] OF THE TRAIL-VIO SYSTEM ON SELF-COLLECTED UAV DATASET.

SEQUENCE	POINT AND LINE FEATURE EXTRACTION / TRACKING	YOLOV8N-SEG INFERENCE	TEMPORAL OBSERVATION RELIABILITY ESTIMATION	RELIABILITY-AWARE POINT-LINE ASSOCIATION AND LOCAL STRUCTURE PRESERVATION	BACK-END OPTIMIZATION AND RELIABILITY-GUIDED MARGINALIZATION	TOTAL RUNTIME
Test01	38.76	30.18	3.42	8.71	35.66	116.73
Test02	31.84	27.36	2.95	6.83	27.42	96.40
Test03	35.29	28.11	3.18	7.52	31.36	105.46
Test04	43.67	31.74	3.89	9.34	42.15	130.79
Test05	40.52	29.63	3.61	8.97	38.64	121.37
Average	38.02	29.40	3.41	8.27	35.05	114.15

As shown in Table IX, the average total runtime of TRAIL-VIO is 114.15 ms per frame, corresponding to approximately 8.76 FPS. The runtime varies across different UAV sequences. This variation is mainly related to the number of extracted and tracked point-line features, the density of valid residuals, keyframe insertion, and the scale of sliding-window optimization.

The current implementation has not yet been fully optimized for strict onboard real-time deployment. Further acceleration may be achieved by TensorRT-based semantic inference, reducing the frequency of semantic updates, optimizing line feature extraction, and adopting an asynchronous front-end pipeline.

VI. CONCLUSION

This study presents TRAIL-VIO, a reliability-aware point-line visual-inertial odometry framework for dynamic and low-texture environments. The proposed method models visual observation quality as a temporally updated reliability state rather than making frame-wise binary decisions. By combining semantic priors with IMU-based motion consistency, TRAIL-VIO suppresses transiently degraded observations caused by motion blur, occlusion, weak texture, and moving objects. The estimated reliability is further used in point-line association, local line-structure preservation, optimization, and marginalization, allowing the system to retain useful structural information while reducing the influence of unreliable visual constraints. Experiments on the EuRoC MAV dataset and the self-collected UAV dataset show that TRAIL-VIO generally achieves more accurate and stable trajectories than the compared methods, especially under rapid motion, dynamic interference, and weak visual texture. The additional analyses further support the role of the proposed modules and show that the framework remains reasonably stable under moderate parameter variations, with most of the extra runtime arising from semantic segmentation. Despite these improvements, several practical limitations remain. The method still depends on semantic segmentation quality. Although the sensitivity analysis shows that the framework can tolerate perturbations of up to $\pm 30\%$ in the initial semantic priors, persistent semantic errors may still degrade the estimation, especially when dynamic objects are repeatedly assigned high reliability. The reliability-related parameters are also manually configured. The system maintains relatively stable tracking when $\tau_{\mathrm{drop}} \in [0.15, 0.35]$ and $k \in [5, 15]$, but these ranges may shift for sensors with different noise characteristics, frame rates, or motion patterns. In addition, the point-line front end still requires sufficient valid geometric constraints. In structure-poor scenes, especially under sustained aggressive rotations, the loss of reliable visual updates may leave IMU preintegration insufficiently corrected and lead to rapid drift. Future work will focus on adaptive reliability estimation, efficient semantic inference, and tighter visual-inertial coupling for more robust onboard deployment.

ACKNOWLEDGMENT

This work is supported by the National Natural Science Foundation of China (NSFC) Joint Fund of Civil Aviation Research under grant U2333214, the Sichuan Science and Technology Program under grant 2026YFHZ0275, the Sichuan Flight Engineering Technology Research Center Project under grant GY2024-11C and the Civil Aviation Administration of China Safety Capacity Building Project under grant MHAQ2024033.

REFERENCES

[1] 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, 2018.

[2] C. Campos, R. Elvira, J. J. G. Rodríguez, J. M. M. Montiel, and J. D. Tardós, "ORB-SLAM3: An accurate open-source library for visual, visual-inertial, and multi-map SLAM," *IEEE Trans. Robot.*, vol. 37, no. 6, pp. 1874–1890, 2021.

[3] Y. He, J. Zhao, Y. Guo, W. He, and K. Yuan, "PL-VIO: Tightly-coupled monocular visual-inertial odometry using point and line features," *Sensors*, vol. 18, no. 4, p. 1159, 2018.

[4] B. Bescos, J. M. Fácil, J. Civera, and J. Neira, "DynaSLAM: Tracking, mapping, and inpainting in dynamic scenes," *IEEE Robot. Autom. Lett.*, vol. 3, no. 4, pp. 4076–4083, 2018.

[5] S. Song, H. Lim, A. J. Lee, and H. Myung, "DynaVINS: A visual-inertial SLAM for dynamic environments," *IEEE Robot. Autom. Lett.*, vol. 7, no. 4, pp. 11523–11530, 2022.

[6] Y. Sun, Q. Wang, C. Yan, Y. Feng, R. Tan, X. Shi, and X. Wang, "D-VINS: Dynamic adaptive visual-inertial SLAM with IMU prior and semantic constraints in dynamic scenes," *Remote Sens.*, vol. 15, no. 15, p. 3881, 2023.

[7] L. von Stumberg and D. Cremers, "DM-VIO: Delayed marginalization visual-inertial odometry," *IEEE Robot. Autom. Lett.*, vol. 7, no. 2, pp. 1408–1415, 2022.

[8] A. I. Mourikis and S. I. Roumeliotis, "A multi-state constraint Kalman filter for vision-aided inertial navigation," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, 2007, pp. 3565–3572.

[9] K. Sun, B. Mohta, K. Pfrommer, M. Watterson, S. Liu, Y. Mulgaonkar, C. J. Taylor, and V. Kumar, "Robust stereo visual inertial odometry for fast autonomous flight," *IEEE Robot. Autom. Lett.*, vol. 3, no. 2, pp. 965–972, 2018.

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

[11] S. Leutenegger, P. Furgale, V. Rabaud, M. Chli, K. Konolige, and R. Siegwart, "Keyframe-based visual-inertial SLAM using nonlinear optimization," in *Proc. Robot. Sci. Syst. (RSS)*, 2013.

[12] P. Geneva, K. Eckenhoff, W. Lee, Y. Yang, and G. Huang, "OpenVINS: A research platform for visual-inertial estimation," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, 2020, pp. 4666–4672.

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

[14] A. Pumarola, A. Vakhitov, A. Agudo, A. Sanfeliu, and F. Moreno-Noguer, "PL-SLAM: Real-time monocular visual SLAM with points and lines," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, 2017, pp. 4503–4508.

[15] Q. Fu, J. Wang, H. Yu, I. Ali, F. Guo, Y. He, and H. Zhang, "PL-VINS: Real-time monocular visual-inertial SLAM with point and line features," *arXiv preprint arXiv:2009.07462*, 2020.

[16] F. Zheng, G. Tsai, Z. Zhang, S. Liu, C.-C. Chu, and H. Hu, "Trifo-VIO: Robust and efficient stereo visual inertial odometry using points and lines," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, 2018, pp. 3686–3693.

[17] Z. Zhao, T. Song, B. Xing, Y. Lei, and Z. Wang, "PLI-VINS: Visual-inertial SLAM based on point-line feature fusion in indoor environment," *Sensors*, vol. 22, no. 14, p. 5457, 2022.

[18] F. Zheng, L. Zhou, W. Lin, J. Liu, and L. Sun, "LRPL-VIO: A lightweight and robust visual-inertial odometry with point and line features," *Sensors*, vol. 24, no. 4, p. 1322, 2024.

[19] J. He, M. Li, Y. Wang, and H. Wang, "PLE-SLAM: A visual-inertial SLAM based on point-line features and efficient IMU initialization," *IEEE Sensors J.*, vol. 25, no. 4, pp. 6801–6811, 2025.

[20] Y. Zhang, P. Zhu, and W. Ren, "PL-CVIO: Point-line cooperative visual-inertial odometry," in *Proc. IEEE Conf. Control Technol. Appl. (CCTA)*, 2023, pp. 859–865.

[21] D. Zou, Y. Wu, L. Pei, H. Ling, and W. Yu, "StructVIO: Visual-inertial odometry with structural regularity of man-made environments," *IEEE Trans. Robot.*, vol. 35, no. 4, pp. 999–1013, 2019.

[22] S. Yang, Y. Song, M. Kaess, and S. Scherer, "Pop-up SLAM: Semantic monocular plane SLAM for low-texture environments," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, 2016, pp. 1222–1229.

[23] 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 *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, 2018, pp. 1168–1174.

[24] B. Bescos, C. Campos, J. D. Tardós, and J. Neira, "DynaSLAM II: Tightly-coupled multi-object tracking and SLAM," *IEEE Robot. Autom. Lett.*, vol. 6, no. 3, pp. 5191–5198, 2021.

[25] L. Cui and C. Ma, "SOF-SLAM: A semantic visual SLAM for dynamic environments," *IEEE Access*, vol. 7, pp. 166528–166539, 2019.

[26] Y. Ren, B. Xu, C. L. Choi, and S. Leutenegger, "Visual-inertial multi-instance dynamic SLAM with object-level relocalisation," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, 2022, pp. 11055–11062.

[27] D. Fu, H. Xia, Y. Liu, and Y. Qiao, "VINS-DIMC: A visual-inertial navigation system for dynamic environment integrating multiple constraints," *ISPRS Int. J. Geo-Inf.*, vol. 11, no. 2, p. 95, 2022.

[28] A. Samadzadeh and A. Nickabadi, "SRVIO: Super robust visual inertial odometry for dynamic environments and challenging loop-closure conditions," *IEEE Trans. Robot.*, vol. 39, no. 4, pp. 2878–2891, 2023.

[29] J. Li, X. Pan, G. Huang, Z. Zhang, N. Wang, H. Bao, and G. Zhang, "RD-VIO: Robust visual-inertial odometry for mobile augmented reality in dynamic environments," *IEEE Trans. Vis. Comput. Graph.*, vol. 30, no. 10, pp. 6941–6955, 2024.

[30] R. Zhou, J. Liu, J. Xie, J. Zhang, Y. Hu, and J. Zhao, "GMS-VINS: Multi-category dynamic objects semantic segmentation for enhanced visual-inertial odometry using a promptable foundation model," *arXiv preprint arXiv:2411.19289*, 2024.

[31] G. Jocher, A. Chaurasia, and J. Qiu, "Ultralytics YOLOv8," GitHub repository, version 8.0.0, 2023. [Online]. Available: https://github.com/ultralytics/ultralytics

[32] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: Common objects in context," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2014, pp. 740–755.

[33] M. Burri, J. Nikolic, P. Gohl, T. Schneider, J. Rehder, S. Omari, M. W. Achtelik, and R. Siegwart, "The EuRoC micro aerial vehicle datasets," *Int. J. Robot. Res.*, vol. 35, no. 10, pp. 1157–1163, 2016.