Multi-sensor Fusion SLAM from the Nadir View for UAV Localization and Mapping

Yawen Li[1], George Vosselman[1], Francesco Nex[1]

[1] Faculty of Geo-Information Science and Earth Observation (ITC), University of Twente, Enschede, The Netherlands
(yawen.li, george.vosselman, f.nex)@utwente.nl

Keywords: UAV, Multi-sensor fusion SLAM, Nadir view, State estimation, Factor graph

Abstract

A single sensor on unmanned aerial vehicles (UAVs) cannot provide stable and accurate trajectory prediction in outdoor low-altitude environments. Moreover, most UAV datasets primarily focus on the low-altitude forward-facing view, with limited coverage of the nadir view. To solve this problem, this study presents a multi-sensor fusion SLAM for UAV localization and mapping from the nadir view in real-time. This method integrates monocular images, IMU measurements, and GNSS coordinates, combining the advantages of each sensor to achieve accurate and reliable state estimation. First, the sensors are initialized and aligned to ensure a consistent reference frame. Subsequently, tracking and local mapping are conducted to establish the system's midterm stability. Finally, the optimization function is formulated using a factor graph that integrates visual factor, inertial factor, GNSS factor, keyframe proximity factor, and designed yaw factor. The system is evaluated using the MARS dataset, and the experimental results demonstrate improved drift reduction and enhanced positioning accuracy.

1. Introduction

Real-time positioning of UAVs can be performed with the help of simultaneous localization and mapping (SLAM) technology to facilitate situational awareness (Huang et al., 2020). Among various SLAM approaches, visual SLAM (VSLAM) has gained significant attention due to its reliance on camera-based perception. Depending on different types of cameras, VSLAM has evolved into monocular SLAM (mono SLAM), when using only image inputs, RGB-D SLAM, and stereo SLAM (Jiang et al., 2021). Compared to depth or stereo cameras, monocular cameras mounted on UAVs are small, flexible, and can perform longer missions. They are more economical, do not require a fixed baseline, and can easily be combined with other sensors. Considering these advantages, mono SLAM serves as a practical technique for UAV positioning in complex environments.

However, relying solely on mono SLAM for positioning is insufficient. It inherently suffers from scale ambiguity (Mur-Artal and Tardós, 2017b). In addition, mono SLAM is susceptible to illumination changes, lack of texture, and dynamic environments, resulting in reduced robustness. Multi-sensor fusion has emerged as an effective solution to these challenges. By integrating sensors such as depth cameras, inertial measurement unit (IMU), global navigation satellite system (GNSS), and laser radar (LiDAR), more robust pose estimation can be provided to compensate for the scale drift of mono SLAM (Li et al., 2023).

Multi-sensor SLAM benefits from the fusion of diverse data sources to boost localization and mapping accuracy, but it still encounters many challenges. These include time synchronization of various sensors, optimization of data fusion strategies, and generalization capabilities in different application scenarios. Addressing these challenges requires accurate temporal synchronization, robust and efficient sensor fusion algorithms, and adaptive optimization strategies tailored to diverse environments.

In this paper, we fuse the visual, inertial, and GNSS measurements to build a tightly-coupled Mono-Inertial-GNSS SLAM system. The work of Cremona et al. (Cremona et al., 2023) that constructed a GNSS-Stereo-Inertial fusion framework by introducing GNSS factors into the ORB-SLAM3 (Campos et al., 2021) system, was also of inspiration for our work. To enhance the versatility of the algorithm, our work employs a monocular camera for experiments. We propose an improved initialization method for the system by adding the GNSS information. In addition to optimizing temporally adjacent keyframes as commonly done in SLAM, we also incorporate spatially nearby keyframes—those that are close in 3D space but not necessarily adjacent in time. This spatial proximity-based selection improves both local and global consistency during optimization. Additionally, a designed yaw constraint is introduced to regulate the UAV's motion within the horizontal plane. The effectiveness of the proposed method is assessed using public datasets. The experimental results show that the proposed improved method has higher accuracy and robustness than ORB-SLAM3 (Mono-Inertial-GNSS).

This study offers the following key contributions:

1. A Mono-Inertial-GNSS framework is proposed for UAV localization and mapping from the nadir view in real-time, rather than the commonly used forward view. All sensors are precisely time-synchronized and integrated within a unified global reference frame, ensuring consistent and accurate multi-sensor fusion.

2. To improve positioning accuracy, the data fusion optimization framework is extended to integrate visual factor, inertial factor, GNSS factor, yaw factor and keyframe proximity factor that are spatially close to the current keyframe.

3. The proposed algorithm is verified in multi-sensor fusion datasets called MARS Dataset using the nadir view. The results prove its effectiveness and superiority in challenging environments, indicating the versatility of our algorithm.

The remainder of this paper is organized as follows. Section II describes the related literature on multi-sensor SLAM. Section

This contribution has been peer-reviewed. The double-blind peer-review was conducted on the basis of the full paper.
https://doi.org/10.5194/isprs-annals-X-2-W2-2025-109-2025

III presents the methodology of the proposed framework. Section IV discusses the experimental results and analysis. Finally, Section V presents the conclusions and future work.

2. Related Works

VSLAM methods can be divided into indirect and direct approaches depending on how image information is processed (Cheng et al., 2022). Indirect (feature-based) methods, such as ORB-SLAM2 (Mur-Artal and Tardós, 2017a), ORB-SLAM3 (Campos et al., 2021), and VINS-Mono (Qin et al., 2018), detect and match features between adjacent frames to estimate camera motion and minimize the re-projection error for accurate pose estimation. In contrast, direct methods, including LSD-SLAM (Engel et al., 2014) and DSO (Engel et al., 2017), infer camera motion using pixel intensity patterns in images without explicitly extracting features.

Among the VSLAM methods, mono SLAM relies on a single camera, making it cost-effective and easy to calibrate. However, mono SLAM struggles to recover true scale due to the absence of depth information and becomes unreliable in fast-motion or low-texture environments.

To overcome these limitations, Visual-Inertial SLAM (VI-SLAM) integrates IMU data to provide additional motion constraints that enhance scale observability, robustness, and localization accuracy. VI-SLAM combines cameras and an IMU to estimate motion states and reconstruct the surrounding environment (Song et al., 2024). By leveraging visual and inertial data, VI-SLAM delivers high-frequency, continuous relative positioning without dependence on external references. However, the absence of a global reference prevents VI-SLAM from directly determining the absolute position, making it prone to cumulative errors. Currently, many researchers are exploring the integration of GNSS into VI-SLAM systems to reduce drift. GNSS offers precise geolocalization in the global earth frame without accumulating errors over time by providing absolute measurement (Cao et al., 2022).

The multi-sensor fusion approach can leverage sensor complementarity to enhance accuracy. To combine these sensors information, there are two sensor fusion coupling methods: loosely-coupled and tightly-coupled (Wang et al., 2024). In the first approach, each sensor's data is processed separately to estimate its position. The final pose is derived through the fusion of estimates from multiple sensors, employing algorithms such as extended Kalman filter (EKF), particle filter, and unscented Kalman filter (UKF) (Lee et al., 2020). The tightly-coupled approach integrates data from multiple sensors to formulate motion and observation equations, enabling joint state estimation with a unified optimization objective (Cadena et al., 2016).

Shen et al. proposed a loosely-coupled UKF framework (Shen et al., 2014) that integrates multiple sensors, such as IMU and a GNSS receiver for position estimation in several scenarios. Based on VINS-Mono (Qin et al., 2018), Qin et al. integrated GNSS into the global estimator to propose VINS-Fusion (Qin et al., 2019). GNSS facilitated the estimation of IMU biases. However, it's a decoupled method for the GNSS and VINS estimators. Yu et al. extended VINS-mono with tightly integrated visual and inertial information and presented a GNSS-aided visual-inertial framework (Yu et al., 2019). Additionally, GNSS measurements were included using a loosely-coupled

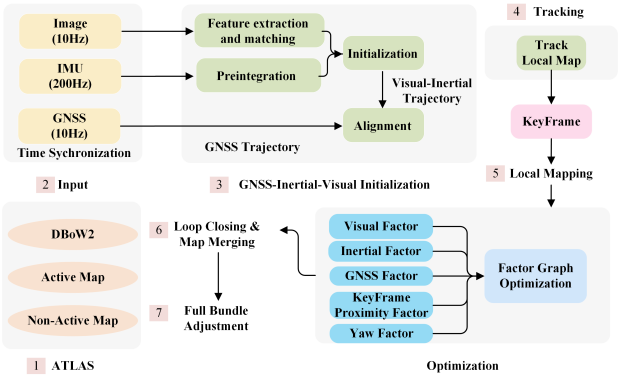


Figure 1. Overview of Mono-Inertial-GNSS framework.

approach. Cao et al. presented GVINS (Cao et al., 2022), a system that tightly integrated GNSS raw measurements into VINS for state estimation. It facilitated global localization in diverse environments, including both indoor and outdoor settings. A probabilistic factor graph framework was used to model the system. The visual and inertial constraints were integrated alongside the incorporation of GNSS pseudorange and doppler shift measurements in the model. Although GVINS builds upon the foundation of VINS-Mono, it is note worthy that no enhancements were made to the visual processing component. Cremona et al. proposed a tightly-coupled GNSS-Stereo-inertial SLAM (Cremona et al., 2023) for agriculture work. The framework extended the visual-inertial mode of ORB-SLAM3 with GNSS measurements. Yu et al. proposed a GNSS/IMU/Vision system (Yu et al., 2025). This system combines multi-sensor data through factor graph optimization and introduces two key improvements: an IMU-assisted optical flow method to suppress dynamic effects and excluding distant features to minimize translation and scale errors.

There are also several aerial mapping approaches that utilize multi-sensor fusion to enhance localization and mapping performance. For example, Map2DFusion (Bu et al., 2016) and OpenREALM (Kern et al., 2020) aim to integrate GNSS and visual information for accurate localization. Map2DFusion aligned 2D image-based SLAM trajectories with GNSS maps, which may result in reduced accuracy due to delayed or inconsistent updates. OpenREALM incorporated global positioning for georeferencing. However, it does not tightly integrate GNSS data into the SLAM optimization process.

Many early methods use loosely coupled or filter-based methods, which limit the full utilization of individual sensor constraints. Recent advancements have incorporated raw GNSS measurements into tightly coupled solutions. Furthermore, these methods are limited to the forward view of vehicles and UAVs, with little consideration from a nadir perspective. Our method performs tightly-coupled multi-sensor fusion, integrating GNSS, IMU, and visual data directly within the SLAM backend. By introducing GNSS measurements as direct constraints in the optimization, our system achieves improved localization accuracy and robustness.

3. Methodology

3.1 Overview of Mono-Inertial-GNSS framework

The overview of the proposed Mono-Inertial-GNSS framework is presented in Figure 1, consisting of the following modules:

1) Atlas: Atlas is a subsystem in ORB-SLAM3 (Campos et al., 2021) used to manage and represent multiple maps, supporting multi-map operations and map merging. In our work, we use the original Atlas implementation provided by ORB-SLAM3. This system contains several sub-maps that can be categorized into two types: active maps and non-active maps. The tracking thread uses active maps. New keyframes are added for continuous optimization and growth of active maps. In contrast, the non-active maps denote reserved maps, and they are not used for tracking threads but are still saved in the Atlas. The system builds a keyframe database in the Atlas using DBoW2 (Gálvez-López and Tardos, 2012). This database can realize different functions, such as relocalization, loop closure detection, and map fusion.

2) Input: Images, IMU, and GNSS measurements serve as inputs to the system. To ensure proper synchronization among these data sources, measurements with closely matching timestamps are selected for joint processing. In our implementation, the Robot Operating System (ROS) framework (Quigley et al., 2009) is used, where each sensor publishes its data as a rostopic. Each message published to a rostopic includes a timestamp indicating the exact time the data is captured. By comparing these timestamps, the system aligns the data temporally to ensure accurate sensor fusion.

Since the sensors operate at different sampling rates (e.g., the IMU at 200 Hz, images at 10 Hz, and GNSS at 10 Hz), not every measurement is used directly in the SLAM process. Instead, we associate each image frame with the temporally closest IMU and GNSS measurements within a predefined time window. Figure 2 shows this temporal alignment process. The data within the orange boxes are aligned images, IMU, and GNSS data within the time window.

3) GNSS-Inertial-Visual Initialization: The system first performs visual initialization, which estimates the initial pose and reconstructs the 3D structure of map points using visual data. After that, IMU pre-integration accumulates inertial measurements between consecutive keyframes, providing constraints that are used to estimate velocity, gravity direction, and sensor biases. These steps establish a local visual-inertial coordinate system. Then, this coordinate system needs to be aligned with the GNSS coordinate system to ensure consistency with global positioning. The visual-inertial coordinates to the GNSS coordinate system are converted based on the external parameters. Finally, the rotation matrix is calculated to ensure the accurate alignment of the two coordinate systems. It is illustrated in Figure 3, which will be discussed in more detail in Section 3.3.

4) Tracking: In the tracking thread, the oriented fast and rotated brief (ORB) (Rublee et al., 2011) algorithm is utilized to identify point features. The relationship between images is obtained by matching the point features. The tracking thread performs real-time localization using the active maps and decides whether to create new keyframes. By including the inertial residuals in the optimization, the body velocity and IMU biases are estimated. Upon losing track, the tracking thread attempts to relocalize the current frame in all the maps of the Atlas. If the relocation is successful, the process will continue. Otherwise, the currently active map will be saved as an inactive map, and a new active map will be recreated. In the tracking thread, only information obtained from adjacent frames or keyframes is used, and the current frame's pose is optimized.

5) Local Mapping: The tracking thread provides keyframes to the local mapping module, then it performs local bundle adjustment (BA) and culls keyframes. This module also sends optimized keyframes to the loop closing thread. The visual factor, inertial factor, GNSS factor, keyframe proximity frame factor, and designed yaw factor are added to the optimization process for refinement. Each factor contains a residual quantifying the error between the predicted and observed values. The optimization process then adjusts the variable values to minimize the sum of these residuals, weighted by their corresponding uncertainties.

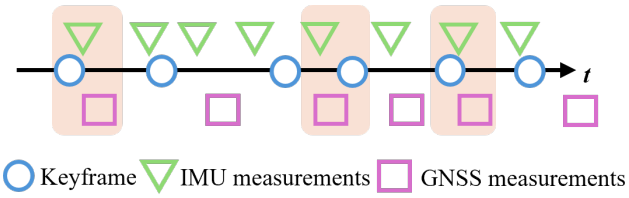


Figure 2. The temporal alignment between different sensor data. The blue circles represent keyframes, green inverted triangles represent IMU measurements, and purple squares represent GNSS measurements. The horizontal axis represents time.

The visual residual is based on the reprojection error between image feature points. The pose and map points are optimized by minimizing the observation error between keyframes and map points. The IMU residual uses a pre-integration model to convert the acceleration and gyroscope velocity information between consecutive frames into pose constraints. GNSS residual provides global pose constraints to reduce cumulative errors. Keyframe proximity residual uses information from spatially adjacent frames to constrain poses. Yaw residual is specifically designed to improve the unobservable yaw ambiguity. A detailed discussion will be provided in Section 3.4. Further, the local mapping adds and deletes keyframes, as well as maps, to the active map, which can reduce the size and amount of local BA. Finally, it constructs a more reliable map and shows an environment more explicitly.

6) Loop closing & map merging: This module is to correct the accumulated pose errors by detecting the loop. In our scenario, the drone takes off and lands at approximately the same location, naturally creating spatial overlap between the beginning and end of the flight. Moreover, the drone follows a strip-based flight trajectory, which results in repeated observations of overlapping areas along adjacent flight lines. These overlaps provide opportunities for loop closure, allowing the system to detect similarities between keyframes recorded at different times and locations. Then, it will optimize the pose and map to eliminate accumulated errors. In addition, the map fusion processes information from multiple maps to provide more comprehensive scene information.

7) Full bundle adjustment: The full bundle adjustment is performed to refine and enhance maps. This module connects multiple sub-maps into an accurate global map with highly accurate pose estimation, which can greatly reduce the overall pose and map errors.

3.2 Frames and Notations

The proposed Mono-Inertial-GNSS framework involves the following coordinate system: the world frame $\{W\}$ is represents the global positions and map points, imu (body) frame $\{B\}$ represents the body frame where the IMU sensor is located, the monocular camera frame $\{C\}$ takes the camera optical center

https://doi.org/10.5194/isprs-annals-X-2-W2-2025-109-2025

as the origin, the GNSS antenna frame $\{G\}$ indicates the installation position of the GNSS antenna and the East-North-Up (ENU) frame $\{E\}$. The $\{W\}$ is a global reference coordinate system defined in the SLAM system, which is first defined by the initialization frame. In this paper, it is recorded as the reference coordinate system of the visual-inertial trajectory (VI trajectory). GNSS outputs latitude, longitude, and altitude values in the geodetic coordinate frame, and these will be transformed into the local Cartesian ENU frame. The first GNSS measurement is typically used as the reference point. The GNSS result in the $\{E\}$ coordinate system is recorded as $\mathbf{e}_i$, i represents the i th keyframe. The transformation between all frames is usually completed through an extrinsic matrix to ensure that all sensor data are optimized and fused in a unified world coordinate system.

3.3 GNSS-Inertial-Visual Initialization

The initial VI-SLAM system has four unobservable directions (Lee et al., 2020). Without external references, the system cannot determine its absolute position and its rotation around the vertical axis. To address the VI-SLAM system initialization problem, the GNSS measurements are incorporated into the visual-inertial initialization process, which estimates the unknown transformation between the VI reference frame and the global frame.

The relative displacement between the current and initial frames is computed using GNSS during monocular camera initialization. The ratio of the GNSS displacement to the camera displacement is obtained as a scale factor. The scale factor is used to adjust the translation part to keep it consistent with the true scale of the GNSS data. Finally, the system updates the frame's pose and creates an initial map to complete the initialization process.

To estimate motion, the system conducts IMU pre-integration on the collected data based on the IMU kinematic model. Unlike traditional IMU kinematic integration, IMU pre-integration accumulates inertial measurements over a time interval to construct relative motion constraints between keyframes. During the nonlinear optimization process, if the initial state (such as velocity, orientation, or bias) changes, traditional integration methods require re-integrating all raw IMU data from the beginning, which is computationally expensive. In contrast, the pre-integration approach processes the IMU data in advance and only applies corrections when certain states are updated, thereby significantly improving optimization efficiency (Forster et al., 2016).

The calculation formulas are as follows:

$$\mathbf{p}_j = \mathbf{p}_i + \mathbf{v}_i \Delta t + \frac{1}{2}\mathbf{R}_i \left(\hat{\mathbf{a}}_i - \mathbf{b}_a\right) \Delta t^2 \tag{1}$$

$$\mathbf{v}_j = \mathbf{v}_i + \mathbf{R}_i \left(\hat{\mathbf{a}}_i - \mathbf{b}_a\right) \Delta t \tag{2}$$

Among them, p_i and p_j represent the position of the start frame and target frame, respectively. The start frame refers to the keyframe at the beginning of the pre-integration interval, while the target frame refers to the keyframe at the end of that interval. v_i and v_j correspond to their velocities of the start frame and target frame. $\hat{\mathbf{a}}_i$ represents the acceleration measurements. b_a represents the accelerometer bias. The rotation matrix of the start frame and the target frame are denoted as R_i and R_j.

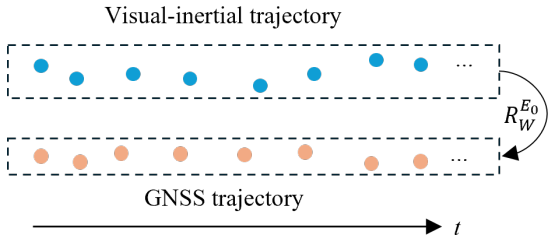


Figure 3. The alignment process.

$$\mathbf{R}_j = \mathbf{R}_i \exp\left(\left(\hat{\omega}_i - \mathbf{b}_g\right) \Delta t\right) \tag{3}$$

$\hat{\omega}_i$ represent angular velocity measurements. b_g represents the gyroscope bias.

The inertial initialization process not only corrects the direction of gravity, but also optimizes the IMU parameters, such as velocity, gravity, and bias.

Figure 3 shows the alignment process. When the visual-inertial initialization process is done, the VI coordinates should be aligned with the global world coordinates. By constraining the time offset between the VI-estimated trajectory and the GNSS trajectory, around 30 poses are selected for matching during the takeoff phase of the drone. Then, the transformation matrix $\mathbf{T}_W^{E_0}$ is calculated, which represents the spatial relationship between the VI coordinate frame and the GNSS coordinate frame. E_0 denotes the coordinate system E evaluated at the initial time (t = 0). The $\mathbf{R}_W^{E_0}$ will be extracted from $\mathbf{T}_W^{E_0}$ for the subsequent alignment and optimization.

3.4 Multi-sensor Fusion

For feature extraction, the images are divided into small grids to realize the distribution of feature points uniformly. To increase the feature extraction rate, the system implements a dynamic adjustment mechanism that decreases the threshold if an insufficient number of features are extracted.

The fusion process is performed within the local bundle adjustment framework, including keyframes in a sliding window along with their corresponding observed 3D points. The sensor state x_i at the i-th time is

$$\mathbf{x}_i = \left[\mathbf{R}_{B_i}^W, \mathbf{t}_{B_i}^W, \mathbf{v}_i^\top, \mathbf{b}_{a_i}^\top, \mathbf{b}_{g_i}^\top\right] \tag{4}$$

where $R_{B_i}^W \in SO(3)$, which represents the rotation matrix of the i th frame with respect to $\{W\}$. The translation of the i-th frame with respect to the world frame W is represented by $t_{B_i}^W \in \mathbb{R}^3$. $v_i \in \mathbb{R}^3$ represents its corresponding velocity. $b_{a_i} \in \mathbb{R}^3$ represents accelerometer bias of the i th frame. $b_{g_i} \in \mathbb{R}^3$ represents gyroscope bias of the i th frame.

$\mathbf{l}_j = \left[X^W, Y^W, Z^W\right]^\top \in \mathbb{R}^3$ are the landmarks in the $\{W\}$. The sensor states within a sliding window denote as $\mathcal{X}_B = [\mathbf{x}_1, \ldots, \mathbf{x}_i, \ldots, \mathbf{x}_N]$. The collection of sensor states spans the most recent N keyframes. Similarly, let

$\mathcal{L} = [\mathbf{l}_1, \ldots, \mathbf{l}_j, \ldots, \mathbf{l}_M]$ denote landmark states observed during these N keyframes. The optimization objective is $\mathcal{X} = \{\mathcal{X}_B, \mathcal{L}\}$.

To solve this optimization problem efficiently, a factor graph is utilized. In this framework, the sensor states and landmarks are treated as variables, and the sensor measurements are encoded as factors that connect the variables. Each factor encodes a residual that measures the prediction error, and optimization seeks variable values that minimize the weighted sum of all such residuals.

The visual residual refers to the distance between the pixel coordinates $\mathbf{u}_{ij}$ obtained by observation and those obtained by projecting the 3D point of j th landmark according to the current estimated pose at i th keyframe. π represents the projection model of the pinhole camera. $\mathbf{T}_C^B$ is the homogeneous transformation matrix from camera frame {C} to body frame {B}. Then, the visual residual $\mathbf{r}_{v_{ij}}$ can be expressed as:

$$\mathbf{r}_{v_{ij}} = \mathbf{u}_{ij} - \Pi\left(\mathbf{T}_C^B \mathbf{T}_B^{W^{-1}} {}^h\mathbf{l}_j\right) \tag{5}$$

where ${}^h\mathbf{l}_j$ is the homogenous form of landmark.

The pre-integration of IMU measurements is carried out between adjacent visual keyframes, denoted by i and $i+1$. The preintegrated measurements of position, rotation, velocity, and covariance are expressed as follows, $\Delta\mathbf{p}_{i,i+1}$, $\Delta\mathbf{R}_{i,i+1}$, $\Delta\mathbf{v}_{i,i+1}$, and $\Sigma_{\mathcal{I}_{i,i+1}}$. They represent accumulated relative rotation, velocity, and position inferred from inertial data captured between the two keyframes. The inertial residual $\mathbf{r}_{\mathcal{I}_{i,i+1}}$ can be expressed as follows:

$$\mathbf{r}_{\mathcal{I}_{i,i+1}} = \left[\mathbf{r}_{\Delta \mathrm{R}_{i,i+1}}, \mathbf{r}_{\Delta \mathbf{v}_{i,i+1}}, \mathbf{r}_{\Delta \mathrm{p}_{i,i+1}}\right] \tag{6}$$

The rotation residual, velocity residual, and position residual can be expressed as follows:

$$\begin{aligned}
\mathbf{r}_{\Delta \mathrm{R}_{i,i+1}} &= \log\left(\Delta\mathbf{R}_{i,i+1}^{\mathrm{T}} \mathbf{R}_i^{\mathrm{T}} \mathbf{R}_{i+1}\right) \\
\mathbf{r}_{\Delta \mathrm{v}_{i,i+1}} &= \mathbf{R}_i^{\mathrm{T}}\left(\mathbf{v}_{i+1} - \mathbf{v}_i - \mathbf{g}\Delta t_{i,i+1}\right) - \Delta\mathbf{v}_{i,i+1} \\
\mathbf{r}_{\Delta \mathrm{p}_{i,i+1}} &= \mathbf{R}_i^{\mathrm{T}}\left(\mathbf{p}_j - \mathbf{p}_i - \mathbf{v}_i\Delta t_{i,i+1} - \frac{1}{2}\mathbf{g}\Delta t^2\right) - \Delta\mathbf{p}_{i,i+1}
\end{aligned} \tag{7}$$

In addition to the visual residual, inertial residual, the GNSS measurement residual is also implemented. $\mathbf{r}_{\mathcal{G}_i}$ is described as:

$$\mathbf{r}_{\mathcal{G}_i} = \mathbf{e}_i - \mathbf{R}_W^{E_0}\left(\mathbf{R}_{B_i}^W \mathbf{t}_E^B + \mathbf{t}_{B_i}^W - \left(\mathbf{R}_{B_0}^W \mathbf{t}_E^B + \mathbf{t}_{B_0}^W\right)\right) \tag{8}$$

where B_0 and E_0 are the coordinate system states of $\{B\}$ and $\{E\}$ at time 0. $\mathbf{R}_W^{E_0}$ is the alignment matrix calculated in section 3.3.

The relative position between keyframes is also taken into account. Let T_{iw}, T_{jw} denote the pose of keyframe KF_i, KF_j. T_{ij} denotes the relative transformation between KF_i and KF_j. Their corresponding GNSS positions are e_i and e_j. The GNSS distance can be expressed as:

$$e_{ij} = \|\mathbf{e}_i - \mathbf{e}_j\| \tag{9}$$

The threshold is set to d_{th}. If e_{ij} exceeds this threshold, both e_i and e_j are discarded.

The relative pose T_{ij} is computed as:

$$T_{ij} = T_{iw}^{-1} \cdot T_{jw} \tag{10}$$

Where T_{iw}^{-1} is the inverse of the pose of KF_i.

Therefore, the keyframe proximity residual $\mathbf{r}_{p_{ij}}$ is defined as:

$$\mathbf{r}_{p_{ij}} = e_{ij} - Tij \tag{11}$$

Since yaw angle errors accumulate over time and affect the trajectory's direction, the designed yaw angle residual is incorporated into the optimization function for correction. The "current orientation" (similar to the yaw) can be estimated by the vector between the current and initial positions as a soft constraint on the yaw during optimization. If the trajectory is relatively straight, the estimated heading will be more stable and can significantly enhance the observability of the yaw. Since we use raw GNSS information, the yaw angle remains unobservable. The designed yaw angle calculated roughly by GNSS is used to constrain the yaw angle estimated by the pose graph. The designed global yaw is defined as:

$$\psi_{\mathrm{gi}} = \arctan\left(\frac{e_{iy}}{e_{ix}}\right) \tag{12}$$

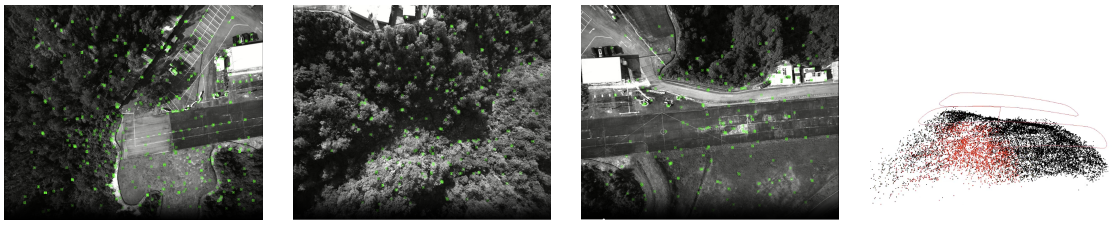
The designed yaw residual $\mathbf{r}_{\psi_i}$ can be expressed:

$$\mathbf{r}_{\psi_i} = \psi_{\mathrm{gi}} - \psi_{\mathrm{esti}} \tag{13}$$

The final optimization cost function is:

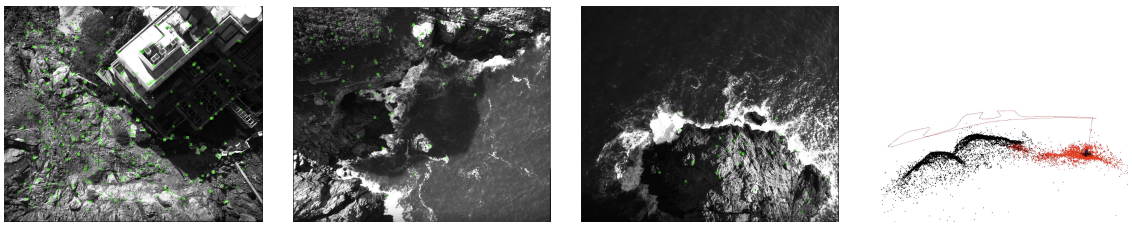
$$\begin{aligned}
\widehat{\mathcal{X}} = \underset{\mathcal{X}}{\operatorname{argmin}}\Bigg(
&\sum_{j=1}^{M}\sum_{i\in\mathcal{K}_j}\rho\left(\left\|\mathbf{r}_{\mathcal{V}_{ij}}\right\|_{\Sigma_{\mathcal{V}_{ij}}^{-1}}\right) \\
&+ \sum_{i=1}^{N}\left\|\mathbf{r}_{\mathcal{I}_{i-1,i}}\right\|^2_{\Sigma_{\mathcal{I}_{i-1,i}}^{-1}} \\
&+ \sum_{i\in\mathcal{N}^*}\rho\left(\left\|\mathbf{r}_{\mathcal{G}_i}\right\|_{\Sigma_{\mathcal{G}_i}^{-1}}\right) \\
&+ \sum_{i\in\mathcal{N}^*}\sum_{j\in K_i\in\mathcal{N}^*}\rho\left(\left\|\mathbf{r}_{\mathcal{P}_{ij}}\right\|_{\Sigma_{\mathcal{P}_{ij}}^{-1}}\right) \\
&+ \sum_{i\in\mathcal{N}^*}\rho\left(\left\|\mathbf{r}_{\psi_i}\right\|_{\Sigma_{\psi_i}^{-1}}\right)\Bigg)
\end{aligned} \tag{14}$$

To maintain real-time performance, our system performs optimization over a sliding window. For the first term, M represents the number of frames within the local sliding window. K_j represents the set of keyframes related to the j th frame. $\rho(\cdot)$ is a robust kernel to suppress outlier effects. Σv_{ij} is the covariance matrix of the visual residual. It is set to represent



(a) (b) (c) (d)

Figure 4. The process of feature extraction and map construction for the airport. (a) Scene 1. (b) Scene 2. (c) Scene 3. (d) Map.



(a) (b) (c) (d)

Figure 5. The process of feature extraction and map construction for the island. (a) Scene 1. (b) Scene 2. (c) Scene 3. (d) Map.

Method	Airport			Island		
	ATE(m)	Drift(%)	Processing time (s)	ATE(m)	Drift(%)	Processing time (s)
ORB-SLAM3 (Mono-Inertial-GNSS)	1.3287	0.065	476.375	1.0256	0.056	477.253
Our Method 1	1.1056	0.054	474.458	0.9152	0.050	476.189
Our Method 2	0.8945	0.044	474.724	0.7582	0.041	476.693

Table 1. The result of MARS Dataset

1-pixel isotropic observation noise. For the second term, N is the sum of IMU samples. $\Sigma_{\mathcal{I}_{i-1,i}}$ is the covariance matrix of the inertial residual. It is obtained by propagating the sensor noise model during the pre-integration process. For the third term, $\mathcal{N}^*$ is a collection of keyframe indexes with GNSS coordinates. $\Sigma_{\mathcal{G}_i}$ is the covariance matrix of the GNSS residual. $\Sigma_{\mathcal{G}_i}$ is set according to the position accuracy in each direction. For the fourth term, $\mathcal{K}_i$ is a set of adjacent keyframes that have relative pose constraints with keyframe i and satisfy: $\mathcal{K}_i = \left\{j \mid \left\|\mathbf{r}_{\mathcal{P}_{ij}}\right\| < d_{th}\right\}$. $\Sigma_{\mathcal{P}_{ij}}$ is the covariance matrix of the keyframe proximity residual. For the fifth term, Σ_{ψ_i} is the covariance matrix of the designed yaw residual, which is set to a constant.

4. Experiments

In this study, ORB-SLAM3 (Mono-Inertial) was modified to ORB-SLAM3 (Mono-Inertial-GNSS) and used as a benchmark for comparison. For comparison, we evaluate three methods: (1) ORB-SLAM3 (Mono-Inertial-GNSS), a baseline method combining monocular visual frames, IMU, and GNSS measurements.; (2) Our method 1 — an extended approach that integrates four types of factors: visual factor, inertial factor, GNSS factor, and a keyframe proximity factor that connects spatially close keyframes to improve local consistency.; and (3) Our method 2 — a further refinement of Method 1, which adds a yaw factor to constrain the heading direction explicitly.

In summary, Our Method 1 includes four residuals that contribute to minimizing errors in the system. Our Method 2, as the final version of our framework, builds upon Method 1 by incorporating the yaw constraint, leading to more stable trajectory estimation.

4.1 Dataset

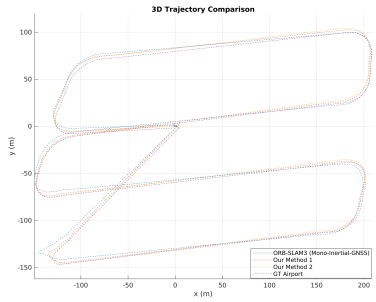
The MARS dataset (Li et al., 2024) is used to evaluate the proposed algorithm. The majority of UAV datasets are designed for frontal or 360-degree perspectives. There are almost no UAV datasets collected from nadir view. MARS dataset makes up for these shortcomings and collects data from the downward-looking view. Data are collected using a DJI M300 RTK quadrotor installed with LIVOX Avia LiDAR, a Hikvision CA-050 RGB camera, and a ZED F9P raw GNSS message receiver. It captures diverse large-area environments, such as an island, a rural town, and a valley. The input data consist of compressed camera images, IMU measurements, and raw GNSS readings. The ground truth is obtained from the high-precision RTK sensor of DJI M300 RTK.

The HKairport_GNSS02 and HKisland_GNSS02 sequences are selected because their corresponding rosbags contain image, IMU, and GNSS rostopics. Additionally, both sequences have suitable cruising altitudes and speeds for our evaluation. The durations of the HKairport_GNSS02 and HKisland_GNSS02 rosbags are 462 seconds and 465 seconds, respectively.

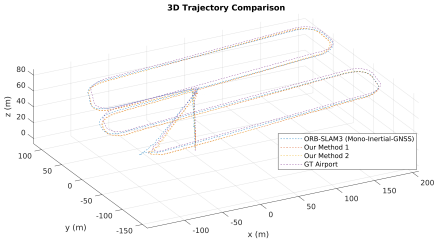
4.2 Evaluation Indicators

The Absolute Trajectory Error (ATE) [unit: m] is used as the evaluation metric. ATE will be calculated by comparing the provided RTK ground truth trajectories with the position trajectories generated by different multi-sensor fusion algorithms. Umeyama's algorithm was applied to register the estimated trajectories to the ground truth (Umeyama, 1991).

ATE serves as a metric for assessing the accuracy and overall consistency of trajectory. Assume that the estimated trajectory

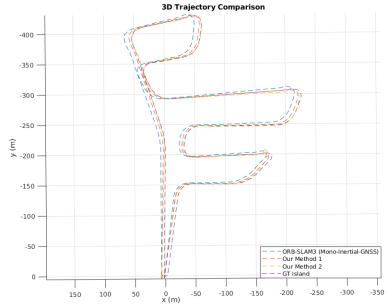


(a)

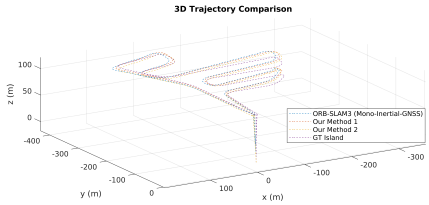


(b)

Figure 6. The trajectory from airport. (a) 2D plane trajectory. (b) 3D trajectory.



(a)



(b)

Figure 7. The trajectory from island. (a) 2D plane trajectory. (b) 3D trajectory.

is denoted by $T_{est,i}$ and the ground truth is denoted by $T_{gt,i}$. Then, $RMSE$ of ATE [unit: m] can be obtained by:

$$ATE = \sqrt{\frac{1}{N}\sum_{i=1}^{N} \left\| \left(T_{gt,i}^{-1}, T_{est,i}\right) \right\|_2^2} \tag{15}$$

The $Drift$ is defined as the ratio of ATE to the total trajectory length. The path lengths of the airport and island are 2.040 km and 1.846 km, respectively.

4.3 Results and Discussion

Figure 4 and Figure 5 show the process of feature extraction and map construction for the airport and island. We randomly selected three scenes from the dataset to showcase the extracted feature points, and the final figure displays both the trajectory and the constructed sparse point maps. The black point cloud represents stable map points, while the red point cloud represents newly observed temporary points in the current frame. The mono-inertial mode of ORB-SLAM3 is prone to failure, and sometimes exhibits significant deviations during turns.

The experiments were performed on a computer equipped with an Intel i7 CPU, NVIDIA GeForce RTX2070MxQ GPU (8GB), and 16GB RAM. Figures 6 and 7 show the trajectories of these algorithms. Table 1 lists the ATE , $Drift$ and $Processing\ time$ for the airport and island. In the airport sequence, for ATE, Our Method 1 reduces the error by 16.79% compared to ORB-SLAM3 (Mono-Inertial-GNSS), and Method 2 reduces it by 32.68%. In the island sequence, for ATE, Our Method 1 reduces the error by 10.76% compared to ORB-SLAM3 (Mono-Inertial-GNSS), and Our Method 2 reduces it by 26.07%. These results demonstrate that our proposed method achieve significantly better localization accuracy than ORB-SLAM3 (Mono-Inertial-GNSS), highlighting the effectiveness of our multi-sensor fusion strategy. In terms of runtime performance, all methods process the data at near real-time speed. Considering that the $HKairport\_GNSS02$ and $HKisland\_GNSS02$ rosbags have durations of 462 seconds and 465 seconds respectively, and the average processing times are close to these durations, the system demonstrates practical feasibility for real-time applications.

5. Conclusions and Future Work

This paper proposes a Mono-Inertial-GNSS multi-sensor SLAM using monocular camera, imu and GNSS based on ORB-SLAM3, aiming to solve the positioning in the outdoor environment from the nadir view. An efficient initialization and alignment method for the coordinate system is proposed, which can rapidly transform various coordinate systems into a unified frame for subsequent processing. A factor graph is constructed to perform state optimization, incorporating five factors: visual, inertial, GNSS, keyframe proximity, and designed yaw factors. These residuals collectively enhance the optimization process, ensuring robust and accurate state estimation by integrating multi-modal sensor data and spatial-temporal constraints. The proposed algorithm is verified on the MARS dataset and the results show that it can achieve higher positioning accuracy than ORB-SLAM3 (Mono-Inertial-GNSS).

This contribution has been peer-reviewed. The double-blind peer-review was conducted on the basis of the full paper.
https://doi.org/10.5194/isprs-annals-X-2-W2-2025-109-2025 | © Author(s) 2025. CC BY 4.0 License.

In the future, field experiments will be conducted to evaluate the system's performance in real-world conditions. To further improve localization accuracy, RTK-based GNSS positioning will be incorporated into the multi-sensor fusion SLAM.

References

Bu, S., Zhao, Y., Wan, G., Liu, Z., 2016. Map2dfusion: Real-time incremental uav image mosaicing based on monocular slam. *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 4564–4571.

Cadena, C., Carlone, L., Carrillo, H., Latif, Y., Scaramuzza, D., Neira, J., Reid, I., Leonard, J. J., 2016. Past, Present, and Future of Simultaneous Localization and Mapping: Toward the Robust-perception Age. *IEEE Transactions on Robotics*, 32(6), 1309–1332.

Campos, C., Elvira, R., Rodríguez, J. J. G., Montiel, J. M., Tardós, J. D., 2021. ORB-SLAM3: An Accurate Open-Source Library for Visual, Visual–Inertial, and Multimap SLAM. *IEEE Transactions on Robotics*, 37(6), 1874–1890.

Cao, S., Lu, X., Shen, S., 2022. GVINS: Tightly Coupled GNSS–Visual–Inertial Fusion for Smooth and Consistent State Estimation. *IEEE Transactions on Robotics*, 38(4), 2004–2021.

Cheng, J., Zhang, L., Chen, Q., Hu, X., Cai, J., 2022. A Review of Visual SLAM Methods for Autonomous Driving Vehicles. *Engineering Applications of Artificial Intelligence*, 114, 104992.

Cremona, J., Civera, J., Kofman, E., Pire, T., 2023. GNSS-stereo-inertial SLAM for Arable Farming. *Journal of Field Robotics*, 41(7), 2215–2225.

Engel, J., Koltun, V., Cremers, D., 2017. Direct Sparse Odometry. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(3), 611–625.

Engel, J., Schoeps, T., Cremers, D., 2014. LSD-SLAM: Large-Scale Direct Monocular SLAM. *Computer Vision - ECCV 2014, Pt Ii*, 8690, 834–849.

Forster, C., Carlone, L., Dellaert, F., Scaramuzza, D., 2016. On-manifold Preintegration for Real-time Visual–Inertial Odometry. *IEEE Transactions on Robotics*, 33(1), 1–21.

Gálvez-López, D., Tardos, J. D., 2012. Bags of Binary Words for Fast Place Recognition in Image sequences. *IEEE Transactions on Robotics*, 28(5), 1188–1197.

Huang, F., Yang, H., Tan, X., Peng, S., Tao, J., Peng, S., 2020. Fast Reconstruction of 3D point Cloud Model Using Visual SLAM on Embedded UAV Development Platform. *Remote Sensing*, 12(20), 3308.

Jiang, S., Jiang, W., Wang, L., 2021. Unmanned Aerial Vehicle-Based Photogrammetric 3D Mapping: A Survey of Techniques, Applications, and Challenges. *IEEE Geoscience and Remote Sensing Magazine*, 10(2), 135–171.

Kern, A., Bobbe, M., Khedar, Y., Bestmann, U., 2020. Openrealm: Real-time mapping for unmanned aerial vehicles. *2020 International Conference on Unmanned Aircraft Systems (ICUAS)*, IEEE, 902–911.

Lee, W., Eckenhoff, K., Geneva, P., Huang, G., 2020. Intermittent gps-aided vio: Online initialization and calibration. *2020 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 5724–5731.

Li, H., Zou, Y., Chen, N., Lin, J., Liu, X., Xu, W., Zheng, C., Li, R., He, D., Kong, F. et al., 2024. MARS-LVIG dataset: A Multi-sensor Aerial Robots SLAM Dataset for LiDAR-Visual-Inertial-GNSS fusion. *The International Journal of Robotics Research*, 43(8), 1114–1127.

Li, S., Li, X., Wang, H., Zhou, Y., Shen, Z., 2023. Multi-GNSS PPP/INS/Vision/LiDAR Tightly Integrated System for Precise Navigation in Urban Environments. *Information Fusion*, 90, 218–232.

Mur-Artal, R., Tardós, J. D., 2017a. ORB-SLAM2: An Open-Source SLAM System for Monocular, Stereo, and RGB-D Cameras. *IEEE Transactions on Robotics*, 33(5), 1255–1262.

Mur-Artal, R., Tardós, J. D., 2017b. Visual-Inertial Monocular SLAM with Map Reuse. *IEEE Robotics and Automation Letters*, 2(2), 796–803.

Qin, T., Cao, S., Pan, J., Shen, S., 2019. A General Optimization-based Framework for Global Pose Estimation with Multiple Sensors. *arXiv preprint, arXiv:1901.03642*.

Qin, T., Li, P., Shen, S., 2018. VINS-Mono: A Robust and Versatile Monocular Visual-Inertial State Estimator. *IEEE Transactions on Robotics*, 34(4), 1004–1020.

Quigley, M., Conley, K., Gerkey, B., Faust, J., Foote, T., Leibs, J., Wheeler, R., Ng, A. Y. et al., 2009. Ros: an open-source robot operating system. *ICRA workshop on open source software*, Kobe, 5.

Rublee, E., Rabaud, V., Konolige, K., Bradski, G., 2011. Orb: An efficient alternative to sift or surf. *2011 International Conference on Computer Vision (ICCV)*, IEEE, 2564–2571.

Shen, S., Mulgaonkar, Y., Michael, N., Kumar, V., 2014. Multi-sensor fusion for robust autonomous flight in indoor and outdoor environments with a rotorcraft mav. *2014 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 4974–4981.

Song, J., Li, W., Duan, C., Wang, L., Fan, Y., Zhu, X., 2024. An Optimization-Based Indoor–Outdoor Seamless Positioning Method Integrating GNSS RTK, PS, and VIO. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 71(5), 2889–2893.

Umeyama, S., 1991. Least-Squares Estimation of Transformation Parameters Between Two Point Patterns. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 13(04), 376–380.

Wang, X., Li, X., Yu, H., Chang, H., Zhou, Y., Li, S., 2024. GIVL-SLAM: A Robust and High-Precision SLAM System by Tightly Coupled GNSS RTK, Inertial, Vision, and LiDAR. *IEEE/ASME Transactions on Mechatronics*, 1–12.

Yu, J., Fang, H., Zhang, X., Wu, W., He, Y., 2025. Tightly Coupled Gnss/Imu/Vision Integrated System for Positioning in Agricultural Scenarios. *preprint*.

Yu, Y., Gao, W., Liu, C., Shen, S., Liu, M., 2019. A gps-aided omnidirectional visual-inertial state estimator in ubiquitous environments. *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 7750–7755.