Satellite Navigation
https://satellite-navigation.springeropen.com/

ORIGINAL ARTICLE

Open Access

M2C-GVIO: motion manifold constraint aided GNSS-visual-inertial odometry for ground vehicles

Tong Hua[1], Ling Pei[1*], Tao Li[1], Jie Yin[1], Guoqing Liu[1] and Wenxian Yu[1]

Abstract

Visual-Inertial Odometry (VIO) has been developed from Simultaneous Localization and Mapping (SLAM) as a low-cost and versatile sensor fusion approach and attracted increasing attention in ground vehicle positioning. However, VIOs usually have the degraded performance in challenging environments and degenerated motion scenarios. In this paper, we propose a ground vehicle-based VIO algorithm based on the Multi-State Constraint Kalman Filter (MSCKF) framework. Based on a unified motion manifold assumption, we derive the measurement model of manifold constraints, including velocity, rotation, and translation constraints. Then we present a robust filter-based algorithm dedicated to ground vehicles, whose key is the real-time manifold noise estimation and adaptive measurement update. Besides, GNSS position measurements are loosely coupled into our approach, where the transformation between GNSS and VIO frame is optimized online. Finally, we theoretically analyze the system observability matrix and observability measures. Our algorithm is tested on both the simulation test and public datasets including Brno Urban dataset and Kaist Urban dataset. We compare the performance of our algorithm with classical VIO algorithms (MSCKF, VINS-Mono, R-VIO, ORB_SLAM3) and GVIO algorithms (GNSS-MSCKF, VINS-Fusion). The results demonstrate that our algorithm is more robust than other compared algorithms, showing a competitive position accuracy and computational efficiency.

Keywords Sensor fusion, Visual-inertial odometry, Motion manifold constraint

Introduction

Accurate pose estimations are essential for abundant robotic applications, such as autonomous driving (Xiong et al., 2021), human navigation (Li et al., 2020), unmanned drone delivery, automatic inspections, etc. Usually, vehicles can be positioned by carrying the Global Navigation Satellite System (GNSS) module or LiDARs which are also widely applied in robot navigation as useful range sensors (Nüchter et al., 2007). Multi-layer LiDARs are heavy and expensive among many sensors. In comparison, Visual-Inertial Odometry (VIO) is more popular because it uses a small and lightweight sensor package and works well in environments where GNSS signals are rejected (Sun et al., 2018).

Although VIOs generally achieve high accuracy in indoor environments, achieving the same good performance for ground vehicles in outdoor environments like urban areas is difficult. Firstly, outdoor environments typically lack reliable features, and rapidly moving vehicles can present a significant challenge for feature matching. Secondly, considerable noise can be generated while driving due to the ground's unevenness and the vehicle's vibration. In addition, the restricted motion on the ground may suffer from the additional unobservable Degree of Freedom (DOF) (Wu et al., 2017). To address the inadequacy of visual and inertial

*Correspondence:
Ling Pei
ling.pei@sjtu.edu.cn
[1] Shanghai Key Laboratory of Navigation and Location Based Services, Shanghai Jiao Tong University, Shanghai, China

sensors in the ground vehicle environment, additional observations are required to constrain the vehicle state and reduce the divergence of positioning error.

Integrating the GNSS sensor is an alternative sensor fusion scheme for eliminating the accumulated errors (Li et al., 2021). Utilizing the derived positions from GNSS, GNSS-VIO can become fully observable and realize a global drift-free localization. However, the integration with the additional sensor may increase the complexity of the navigation system, especially for a graph optimization-based framework (Gong et al., 2020; Cioffi & Scaramuzza, 2020). Some Kalman filter-based methods like Multi-State Constraint Kalman Filter (MSCKF) (Mourikis et al., 2007) have demonstrated both precision and computational efficiency, thus we focus a filter-based scheme for sensor fusion.

The kinematic constraint is another effective auxiliary update information for improving error estimation based on the fact that the robots are on the ground manifold (Li et al., 2012). It does not require additional sensors and has strong autonomy (Ning et al., 2021). There are various descriptions of kinematic constraints such as the Non-Holonomic Constraint (NHC) and plane constraint, whose models depend on some specific motion manifolds. but they lack a unified representation which can be extended to specific constraints easily. Meanwhile, although the measurement model can be derived theoretically, real world scenes may not satisfy the model assumption. It should be noted that the noise of motion manifold constraints is highly correlated with the vehicle's motion. The model assumption may not be satisfied when the vehicle bumps or turns. Therefore, adaptive filtering is a viable approach to tackling the challenge. Furthermore, its effect on the system observability should be investigated to guarantee that it is beneficial for a new measurement.

This paper focuses on the fusion of VIO, GNSS, and motion manifold constraints to cope with the challenges of accuracy and time efficiency. Our contributions are as follows:

- We provide a GNSS-aided filter-based VIO approach, which introduces multiple measurements including visual measurements, GNSS positions, and motion manifold constraints.
- We provide a unified motion manifold measurement model, including rotation, velocity, and translation constraints, and propose an adaptive filtering algorithm to promote the measurement robustness.
- By defining the local observability matrix, we analyze the impact of multiple constraints on the whole system.

The remaining paper is organized as follows: the second section discusses the related work. Then, we sort out the overall framework of the proposed algorithm, where the motion manifold constraint-based filtering method is detailed. The fourth section analyzes the system's observability in an ideal model. In the evaluation section, our work is compared with different VIOs and GNSS-VIOs in the simulation and urban datasets. The final section concludes the paper.

Related works

Visual-inertial odometry

As a multi-sensor fusion approach, VIO is applied to harsh scenarios such as texture-less, motion blur, and occlusions. In addition, it solves the scale estimation problem of monocular visual Simultaneous Localization and Mapping (SLAM) (Campos et al., 2021b). In general, VIO updates poses by filter-based methods (Ribeiro, 2004; Wan, 2000) or graph optimization-based methods (Grisetti et al., 2011).

Several graph optimization-based VIO algorithms can significantly improve the accuracy. Representative approaches include OKVIS (Leutenegger et al., 2013) and VINS-Mono (Qin et al., 2018), which develop an Inertial Measurement Unit (IMU) pre-integration technique (Forster et al., 2015), and ORB_SLAM3 (Campos et al., 2021b) introduces ORB corner extraction method.

As a tightly coupled approach, MSCKF inherits the filtering framework of EKF and solves the problem of excessive dimension growth in EKF, which has the great advantages of accurate positioning and light weight. MSCKF is further refined by deriving a closed-form IMU state transition equation and applying First Estimate Jacobian (FEJ) for improving consistency (Li & Mourikis, 2013). MSCKF has also extended to a stereo version (Sun et al., 2018), which is applied to Micro Aerial Vehicles (MAVs).

There are several works on GNSS-VIO fusion. Qin et al. (2019) propose VINS-Fusion which is a loosely-coupled estimator fusing GNSS relative poses based on VINS-Mono. Gong et al. (2020); Cioffi and Scaramuzza (2020) adopt a loosely coupled graph optimization-based approach using GNSS position and velocity. Cao et al. (2022), Liu et al. (2021), however, tightly couple raw GNSS data into the visual-inertial system. Li et al. (2022) utilize Precise Point Positioning (PPP) in a factor graph framework and improves the accuracy through high precision carrier phase. Liu et al. propose g-MSCKF and emphasize its observability-aware advantage, while Lee et al. (2022) optimize the spatiotemporal calibration between IMU-GNSS in the proposed GAINS. These

filter-based works show a comparable level of performance compared to optimization-based methods, yet they have not taken into account manifold constraints in challenging environments.

Kinematic constraints in vehicle localization

Pose estimation can be optimized for vehicle localization with the prior constraint information on the vehicle body and the ground. Kinematic constraints are combined in filter-based (Ma et al., 2019) and optimization-based (Yu et al., 2021) VIO based on Ackerman steering model. In LARVIO proposed by Xiaochen et al. (2020), the application of Zero Velocity Update (ZUPT) is judged by the movement of the visual image pixels. ZUPT can prevent the divergence of filtering effectively, while it can only work in the scenario of a stationary vehicle. Tian et al. (2021) recover the scale by optimizing the height difference between the vehicle and the ground. Zhang et al. (2021) reconstruct the full three-dimensional value of angular velocity through mathematical derivation with the constraints of wheel odometer and ground manifold.

Another kinematic constraint, namely NHC, is applicable in Inertial Navigation Systems (INS) for land vehicles (Sukkarieh, 2000; Shin et al., 2002). 3D Auxiliary Velocity Updates (AVUs) encompassing NHC and odometer-derived velocity are used to improve the accuracy when GNSS signals are blocked (Niu et al., 2007). And Zhang et al. (2021) propose a novel algorithm to meet the need for all-wheel steering robot positioning, which extends the application of kinematic constraints. However, these works are often combined with GNSS sensors and are rarely associated with VIO. Apart from NHC, some works are trying to enhance localization accuracy by imposing plane constraints (Wu et al., 2017; Panahandeh et al., 2012), which can correct the system solution effectively. Nevertheless, these kinematic constraints have not been incorporated into a unified manifold representation, and most of them do not consider the adjustment of measurement noise, which has a great influence on the robustness of positioning in real scenarios.

Table 1 Glossary of notations

Symbol	Meaning
G	World frame
N	East-North-Up(ENU) frame
b	Body frame
I	IMU frame
C	Camera frame
${}^{B}_{A}\mathbf{q}({}^{B}_{A}\mathbf{R})$	Rotation from frame A to frame B
${}^{B}\mathbf{v}_{A}$	Frame A's velocity in frame B
${}^{B}\mathbf{p}_{A}$	Frame A's position in frame B
$\hat{\mathbf{x}}$	Estimated value of $\mathbf{x}$
$\tilde{\mathbf{x}}$	Error value of $\mathbf{x}$
$\mathbf{e}_i$	The ith column of $\mathbf{I}_3$

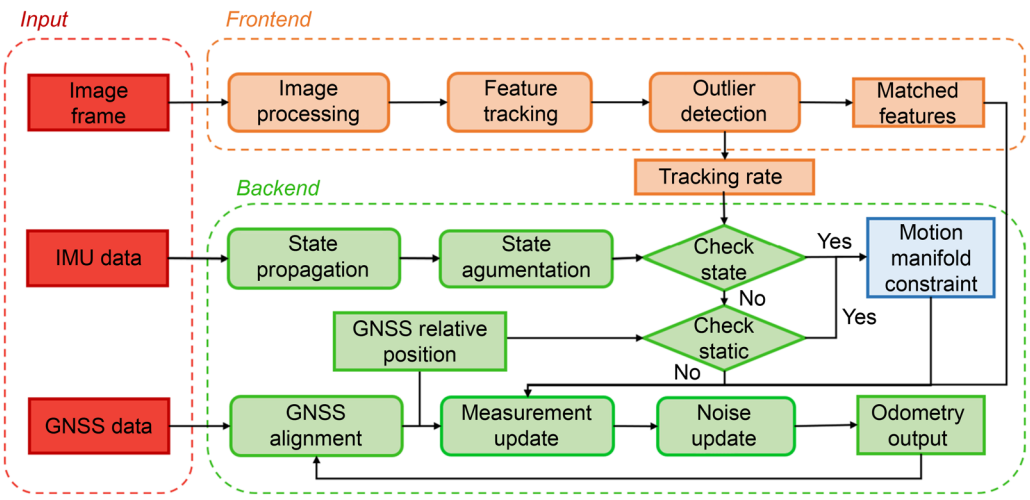

Fig. 1 M2C-GVIO system overview

Filter description

Before describing the overall filtering algorithm, we follow the notation in (Sun et al., 2018). The coordinate frames and some notations involved are clarified in Table 1. The IMU frame and body frame are equivalent by default in the following discussion. Our system overview is illustrated in Fig. 1.

State definition

The state vector of the system is defined as:

$$\begin{cases} \mathbf{X} = \begin{bmatrix} \boldsymbol{X}_{\mathrm{I}}^{\mathrm{T}} & \boldsymbol{X}_{\mathrm{C}_1}^{\mathrm{T}} & \boldsymbol{X}_{\mathrm{C}_2}^{\mathrm{T}} & \cdots & \boldsymbol{X}_{\mathrm{C}_M}^{\mathrm{T}} \end{bmatrix}^T \\ \boldsymbol{X}_{\mathrm{I}} = \begin{bmatrix} \boldsymbol{X}_{\mathrm{i}}^{\mathrm{T}} & \boldsymbol{X}_{e}^{\mathrm{T}} & \boldsymbol{X}_{\mathrm{I}}^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}} \\ \boldsymbol{X}_{\mathrm{C}} = \begin{bmatrix} {}_{\mathrm{G}}^{\mathrm{C}}\boldsymbol{q}^{\mathrm{T}} & {}^{\mathrm{G}}\boldsymbol{p}_{\mathrm{C}}^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}} \\ \boldsymbol{X}_{\mathrm{i}} = \begin{bmatrix} {}_{\mathrm{G}}^{\mathrm{I}}\boldsymbol{q}^{\mathrm{T}} & \boldsymbol{b}_{\mathrm{g}}^{\mathrm{T}} & {}^{\mathrm{G}}\boldsymbol{v}_{\mathrm{I}}^{\mathrm{T}} & \mathbf{b}_{\mathrm{a}}^{\mathrm{T}} & {}^{\mathrm{G}}\boldsymbol{p}_{\mathrm{I}}^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}} \\ \boldsymbol{X}_{\mathrm{e}} = \begin{bmatrix} {}_{\mathrm{C}}^{\mathrm{I}}\boldsymbol{q}^{\mathrm{T}} & {}^{\mathrm{I}}\boldsymbol{p}_{\mathrm{C}}^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}} \\ \boldsymbol{X}_{I} = \begin{bmatrix} {}_{\mathrm{G}}^{\mathrm{N}}\boldsymbol{q}^{T} & {}^{\mathrm{N}}\boldsymbol{p}_{\mathrm{G}}^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}} \end{cases} \tag{1}$$

where ${}_{G}^{I}\mathbf{q}$ is the rotation from the global frame to the IMU frame, $\mathbf{b}_g$ and $\mathbf{b}_a$ are the biases of the measured angular velocity and linear acceleration from IMU. ${}^{G}\mathbf{v}_I$ and ${}^{G}\mathbf{p}_I$ are the velocity and position of the IMU frame in the global frame. ${}_{C}^{I}\mathbf{q}, {}^{I}\mathbf{p}_C$ are the extrinsic parameters for online calibration, and ${}_{G}^{N}\mathbf{q}$ and ${}^{N}\mathbf{p}_G$ represent the transformation from the global frame to the ENU frame. $\mathbf{X}_C$ is the augmented state following (Mourikis et al., 2007), and M is the length of the sliding window. For accuracy and convenience, we introduce the error state vector in the description of the filter:

$$\begin{cases} \tilde{\mathbf{X}}_I = \begin{bmatrix} {}_{G}^{I}\tilde{\boldsymbol{\theta}}^{\mathrm{T}} & \tilde{\boldsymbol{b}}_g^{\mathrm{T}} & {}^{\mathrm{G}}\tilde{\boldsymbol{v}}_{\mathrm{I}}^{\mathrm{T}} & \tilde{\boldsymbol{b}}_{\mathrm{a}}^{\mathrm{T}} & {}^{\mathrm{G}}\tilde{\boldsymbol{p}}_{\mathrm{I}}^{\mathrm{T}} & {}_{\mathrm{C}}^{\mathrm{I}}\tilde{\boldsymbol{\theta}}^{\mathrm{T}} & {}^{\mathrm{I}}\tilde{\boldsymbol{p}}_{\mathrm{C}}^{\mathrm{T}} & {}_{\mathrm{G}}^{\mathrm{N}}\tilde{\boldsymbol{\theta}}^{\mathrm{T}} & {}^{\mathrm{N}}\tilde{\boldsymbol{p}}_{\mathrm{G}}^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}} \\ \tilde{\boldsymbol{X}}_C = \begin{bmatrix} {}_{G}^{C}\tilde{\boldsymbol{\theta}}^{T} & {}_{C}^{G}\boldsymbol{p}^{T} \end{bmatrix}^{T} \end{cases} \tag{2}$$

where $\tilde{\mathbf{X}}_I$ and $\tilde{\mathbf{X}}_C$ are the error IMU state vector and error camera state vector respectively. $\tilde{\boldsymbol{\theta}}$ comes from the first three dimensions of the quaternion:

$$\delta\mathbf{q} = \begin{bmatrix} \frac{1}{2}\tilde{\boldsymbol{\theta}}^{\mathrm{T}} & 1 \end{bmatrix}^{\mathrm{T}} \tag{3}$$

where the quaternion $\delta\mathbf{q}$ describes the small rotation that makes the true and estimated attitude coincide.

State propagation

For a low-cost IMU, the continuous kinematic model is given by Eq. (4) ignoring the earth rotation effect.

$$\begin{cases} {}_{\mathrm{I}}^{\mathrm{G}}\dot{\mathbf{q}} = \frac{1}{2}\boldsymbol{\Omega}(\boldsymbol{\omega}_{\mathrm{m}} - \boldsymbol{b}_{\mathrm{g}} - \boldsymbol{n}_{\mathrm{g}}){}_{\mathrm{I}}^{\mathrm{G}}\boldsymbol{q} \\ {}^{\mathrm{G}}\dot{\mathbf{v}}_{\mathrm{I}} = {}_{\mathrm{I}}^{\mathrm{G}}\boldsymbol{R}(\boldsymbol{a}_{\mathrm{m}} - \boldsymbol{b}_{\mathrm{a}} - \boldsymbol{n}_{\mathrm{a}}) + {}^{\mathrm{G}}\boldsymbol{g} \\ {}^{\mathrm{G}}\dot{\mathbf{p}}_I = {}^{\mathrm{G}}\boldsymbol{v}_{\mathrm{I}} \\ {}_{\mathrm{C}}^{I}\dot{\mathbf{q}} = \mathbf{0}_{3\times1} \\ {}^{\mathrm{I}}\dot{\mathbf{p}}_{\mathrm{C}} = \mathbf{0}_{3\times1} \\ {}_{\mathrm{G}}^{\mathrm{N}}\dot{\mathbf{q}} = \mathbf{0}_{3\times1} \\ {}^{\mathrm{N}}\dot{\mathbf{p}}_G = \mathbf{0}_{3\times1} \\ \dot{\mathbf{b}}_{\mathrm{g}} = \boldsymbol{n}_{wg} \\ \dot{\mathbf{b}}_a = \boldsymbol{n}_{wa} \end{cases} \tag{4}$$

where $\boldsymbol{\omega}_m$ and $\mathbf{a}_m$ are the angular velocity and linear acceleration in the local IMU frame. ${}^{G}\mathbf{g}$ is the gravity. $\mathbf{n}_g$ and $\mathbf{n}_a$ are white Gaussian noises of the gyroscope and accelerometer measurements. $\mathbf{n}_{wg}$ and $\mathbf{n}_{wa}$ are the random walk rates of the gyroscope and accelerometer measurement biases (Sun et al., 2018). $\Omega(\boldsymbol{\omega})$ is defined as:

$$\boldsymbol{\Omega}(\omega) = \begin{bmatrix} -\lfloor\boldsymbol{\omega}\times\rfloor & \boldsymbol{\omega} \\ -\boldsymbol{\omega}^{\mathrm{T}} & 0 \end{bmatrix} \tag{5}$$

After linearizing Eq. (4), we can obtain the error state propagation equation:

$$\dot{\tilde{\mathbf{X}}}_{\mathrm{I}} = \mathbf{F}\tilde{\mathbf{X}}_{\mathrm{I}} + \boldsymbol{G}\boldsymbol{n}_{\mathrm{I}} \tag{6}$$

where $\mathbf{n}_I = \begin{bmatrix} \mathbf{n}_g^T & \mathbf{n}_{wg}^T & \mathbf{n}_a^T & \mathbf{n}_{wa}^T \end{bmatrix}^T$ is the process noise. $\mathbf{F}$ is the continuous state transition matrix and $\mathbf{G}$ is the input noise Jacobian, as shown in Eq. (8). And the state covariance is obtained by the discrete time state transition $\boldsymbol{\Phi}_k$ and noise covariance matrix $\mathbf{Q}_k$ with respect to $\mathbf{F}$ and $\mathbf{G}$. Then the propagated covariance is:

$$\boldsymbol{P}_{k+1|k} = \boldsymbol{\Phi}_k\boldsymbol{P}_k\boldsymbol{\Phi}_k^{\mathrm{T}} + \boldsymbol{Q}_k \tag{7}$$

where $\mathbf{P}_k$ is the state covariance matrix at time step k, and $\mathbf{P}_{k+1|k}$ is the predicted covariance matrix at time step $k+1$ based on the IMU inputs at time step k.

$$\begin{cases} \mathbf{F} = \begin{bmatrix} -\left[\widehat{\boldsymbol{\omega}}_{\times}\right] & -\boldsymbol{I}_3 & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times12} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times12} \\ -{}_{I}^{G}\hat{\boldsymbol{R}}\lfloor\hat{\boldsymbol{a}}_{\times}\rfloor & \mathbf{0}_{3\times3} & \boldsymbol{I}_{3\times3} & -{}_{I}^{G}\hat{\boldsymbol{R}} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times12} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times12} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \boldsymbol{I}_3 & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times12} \\ \mathbf{0}_{12\times3} & \mathbf{0}_{12\times3} & \mathbf{0}_{12\times3} & \mathbf{0}_{12\times3} & \mathbf{0}_{12\times3} & \mathbf{0}_{12\times12} \end{bmatrix} \\ \boldsymbol{G} = \begin{bmatrix} -\boldsymbol{I}_3 & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \boldsymbol{I}_3 & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & -{}_{I}^{G}\hat{\boldsymbol{R}} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \boldsymbol{I}_3 \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{12\times3} & \mathbf{0}_{12\times3} & \mathbf{0}_{12\times3} & \mathbf{0}_{12\times3} \end{bmatrix} \end{cases} \tag{8}$$

State augmentation

When detecting a new image, we add the new camera error state $\mathbf{X}_{C_{M+1}}$ to the sliding window. The covariance matrix is also augmented by the Jacobian matrix $\mathbf{J}$ where:

$$\mathbf{J} = \frac{\partial \boldsymbol{X}_{C_{M+1}}}{\partial \boldsymbol{X}} = \begin{bmatrix} {}^{\mathrm{C}}_{\mathrm{I}}\hat{\boldsymbol{R}} & \mathbf{0}_{3\times 9} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times(12+6M)} \\ -{}^{\mathrm{I}}_{\mathrm{G}}\hat{\boldsymbol{R}}^{\mathrm{T}}\left\lfloor {}^{\mathrm{I}}\hat{\boldsymbol{p}}_{\mathrm{C}\times}\right\rfloor & \mathbf{0}_{3\times 9} & \boldsymbol{I}_3 & \mathbf{0}_{3\times(12+6M)} \end{bmatrix} \tag{9}$$

Visual measurement model

After initializing the three-dimensional coordinates of multiple observed features by the triangulation method, a two-dimensional projection $\mathbf{z}$ of a feature point ${}^{G}\mathbf{p}_f$ on the normalized image plane is given by Eq. (10).

$$\mathbf{z} = \frac{1}{{}^{C}\mathbf{z}_f}\begin{bmatrix} {}^{C}\mathbf{x}_f \\ {}^{C}\mathbf{y}_f \end{bmatrix} = h(\mathbf{X}, {}^{\mathrm{G}}\mathbf{p}_f) \tag{10}$$

where ${}^{C}\mathbf{p}_f = \begin{bmatrix} {}^{G}\mathbf{x}_f & {}^{G}\mathbf{y}_f & {}^{G}\mathbf{z}_f \end{bmatrix}^T$ denotes the feature position in the camera frame. Once the feature projection is estimated, the linear model of reprojection error can be derived:

$$\hat{\boldsymbol{z}} = \boldsymbol{H}_X\tilde{\boldsymbol{X}} + \boldsymbol{H}_f{}^{\mathrm{G}}\boldsymbol{p}_{\mathrm{f}} + \boldsymbol{n} \tag{11}$$

where $\mathbf{n}$ is the observation Gaussian noise. To eliminate the influence of $\mathbf{p}_f$ in measurement, we define a unitary matrix $\mathbf{V}$ whose columns form the basis of the left nullspace of $\mathbf{H}_f$ (Mourikis et al., 2007), and transform Eq. (11) into:

$$\boldsymbol{r}_{\mathrm{C}} = \mathbf{V}^{\mathrm{T}}\tilde{\mathbf{z}} \triangleq \mathbf{H}_{\mathrm{C}}\tilde{\mathbf{X}} + \boldsymbol{n}_o \tag{12}$$

where $\mathbf{r}_c$ is the projected residual. The measurement update of Kalman filter can be performed using Eq. (12).

In the frontend implementation, we adopt the technology of contrast limited adaptive histogram equalization (Zuiderveld, 1994) (CLAHE) to preprocess the image. This method not only enhances the image's contrast so that the algorithm can be applied under weak light conditions but also reduces the noise, which is beneficial for feature matching. Similar methods are applied in (Campos et al., 2021b; Qin et al., 2018). Some VIOs use Fast detector (Sun et al., 2018; Bloesch et al., 2015) to save computing resources. However, feature points such as ORB corners can achieve more accurate estimation if the vehicle has an aggressive motion in the outdoor scene. Finally, the KLT optical flow algorithm (Lucas & Kanade, 1997) and RANSAC method (Fischler & Bolles, 1981) are adopted to track the features and eliminate the outliers.

Motion manifold measurement model

To describe the pose constraints of a robot on the ground, we model the ground as the general motion manifold (Zhang et al., 2021):

$$\mathcal{M}(\mathbf{p}) = 0 \tag{13}$$

where $\mathbf{p}$ is the position of the ground robot. Meanwhile, when the ground robot is on the manifold, the z-axis in the global frame is collinear with the gradient of the motion manifold. Therefore the following equation holds:

$$\left\lfloor \left({}^{G}_{b}\boldsymbol{R} \cdot \boldsymbol{e}_3 \right)_{\times} \right\rfloor \cdot \nabla\mathcal{M}({}^{G}\boldsymbol{p}_{\mathrm{b}}) = 0 \tag{14}$$

Based on the above definitions and assumptions, some state constraints can be obtained.

Velocity constraints

When the ground vehicle is in close contact with the ground, the following equations can be derive from Eqs. (13) and (14):

$$\begin{cases} \dfrac{\partial\mathcal{M}({}^{\mathrm{G}}\boldsymbol{p}_b)}{\partial t} = 0 \\ \nabla\mathcal{M}({}^{\mathrm{G}}\boldsymbol{p}_b) \cdot {}^{\mathrm{G}}\boldsymbol{v}_b = 0 \\ \boldsymbol{e}_3^{\mathrm{T}} \cdot {}^{\mathrm{G}}_{b}\boldsymbol{R} \cdot {}^{\mathrm{G}}\boldsymbol{v}_b = 0 \\ {}^{\mathrm{b}}v_z = 0 \end{cases} \tag{15}$$

A similar equation ${}^{b}v_y = 0$ holds since the vehicle is also on the motion manifold in the cross-track direction (y-axis). Assuming that the constraint in Eq. (15) is stochastic because of the deviation in the real world, the velocity measurement model can be obtained:

$$\begin{cases} \boldsymbol{r}_{vel} = \mathbf{0}_{2\times 1} - \begin{bmatrix} {}^{\mathrm{b}}\boldsymbol{v}_y \\ {}^{\mathrm{b}}\boldsymbol{v}_z \end{bmatrix} = \boldsymbol{H}_{vel}\tilde{\mathbf{X}} + \boldsymbol{n}_n \\ \boldsymbol{H}_{vel} = \boldsymbol{\Lambda}_1\left[\left\lfloor {}^{\mathrm{b}}\hat{\mathbf{v}}_{\times} \right\rfloor \mathbf{0}_{3\times 3}\ {}^{\mathrm{b}}_{G}\hat{\mathbf{R}}\ \mathbf{0}_{3\times(18+6M)} \right] \end{cases} \tag{16}$$

where $\mathbf{r}_{vel}$ and $\mathbf{H}_{vel}$ are the residual and measurement Jacobian, respectively. $\boldsymbol{\Lambda}_1 = \begin{bmatrix} \mathbf{e}_2 & \mathbf{e}_3 \end{bmatrix}^T$, $\mathbf{n}_n$ is the observation noise, and the body frame is equivalent to the IMU frame. The noise setting of velocity constraints depends on the current motion, which will be elaborated in the adaptive filtering section.

A special scenario is that when there is no specific motion being detected which means $\mathbf{p}$ is constant in Eq. (13), and its first derivative (velocity) and second derivative (acceleration) should be zero. Thus we apply Eq. (17) as a pseudo-measurement of robot velocity and IMU acceleration bias:

$$\boldsymbol{r}_z = \begin{bmatrix} \mathbf{0}_{3\times1} \\ \boldsymbol{a}_{\mathrm{m}} \end{bmatrix} - \begin{bmatrix} {}^{\mathrm{G}}\boldsymbol{v}_I \\ \boldsymbol{b}_{\mathrm{a}} - {}^{\mathrm{I}}_{\mathrm{G}}\boldsymbol{R}^{G}\boldsymbol{g} \end{bmatrix} \tag{17}$$

where $\mathbf{r}_z$ is the velocity and acceleration bias residual, and $\mathbf{a}_m$ is the IMU acceleration measurement. Following Eq. (17), the corresponding measurement Jacobians are given by:

$$\boldsymbol{H}_z = \begin{bmatrix} \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \boldsymbol{I}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times(15+6M)} \\ \left\lfloor ({}^{\mathrm{I}}_{G}\hat{\mathbf{R}}^{\mathrm{G}}\boldsymbol{g})_{\times} \right\rfloor & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \boldsymbol{I}_{3\times3} & \mathbf{0}_{3\times(15+6M)} \end{bmatrix} \tag{18}$$

In the proposed scheme, we detect the static scene by the average moving pixel distance of features and the relative displacement of GNSS position measurements between two image frames. The constraint starts only when the calculated values are less than certain thresholds such as 0.1 pixels for visual features or 0.005 m for GNSS measurements, which depend on the image size and the noise of GNSS measurements.

Rotation & translation constraints

Assuming that the vehicle is running on the plane π (Wu et al., 2017), the motion manifold can be specified as:

$$\mathrm{M}(\boldsymbol{p}) = a_0 + a_1 p_x + a_2 p_y + a_3 p_z = a_0 + \boldsymbol{A}\boldsymbol{p} = \mathbf{0} \tag{19}$$

where $\mathbf{A} = \begin{bmatrix} a_1 & a_2 & a_3 \end{bmatrix}$. Furthermore, if π is set as the initial x-y plane of the global frame ($a_0 = a_1 = a_2 = 0$), the following equations hold:

$$\begin{cases} {}^{\mathrm{G}}_{I}\boldsymbol{R} \cdot \boldsymbol{e}_3 = k \cdot [a_1 \ a_2 \ a_3]^{\mathrm{T}} \Rightarrow \boldsymbol{\Lambda}_2 {}^{\mathrm{G}}_{I}\boldsymbol{R} \cdot \boldsymbol{e}_3 = \mathbf{0} \\ {}^{\mathrm{G}}p_{I,z} = 0 \end{cases} \tag{20}$$

where $\Lambda_2 = \begin{bmatrix} \mathbf{e}_1 & \mathbf{e}_2 \end{bmatrix}^T$. Thus a rotational roll-pitch constraint and a translation constraint on the z-axis can be obtained:

$$\boldsymbol{r}_{rot} = \mathbf{0}_{2\times1} - \boldsymbol{z}_{rot} = -\boldsymbol{\Lambda}_2 {}^{\pi}_{G}\boldsymbol{R}^{\mathrm{G}}_{I}\boldsymbol{R} \cdot \boldsymbol{e}_3 \tag{21}$$

$$r_{tran} = 0 - z_{tran} = -\boldsymbol{e}_3^{\mathrm{T}} {}^{\pi}_{G}\boldsymbol{R}^{\mathrm{G}}\boldsymbol{p}_{\mathrm{I}} \tag{22}$$

where $\mathbf{z}_{rot}$ is the estimated roll and pitch, and z_{tran} is the estimated translation on the z-axis. The corresponding measurement Jacobians are given by:

$$\begin{cases} \boldsymbol{H}_{rot} = \boldsymbol{\Lambda}_2 \left[-{}^{\pi}_{G}\boldsymbol{R}^{\mathrm{G}}_{I}\hat{\boldsymbol{R}} \lfloor \boldsymbol{e}_3 \times \rfloor \ \mathbf{0}_{3\times(24+6M)} \right] \\ \boldsymbol{H}_{tran} = \boldsymbol{e}_3^{\mathrm{T}} \left[\mathbf{0}_{3\times12} \ {}^{\pi}_{G}\boldsymbol{R} \ \mathbf{0}_{3\times(24+6M)} \right] \end{cases} \tag{23}$$

where ${}^{\pi}_{G}\mathbf{R}$ is an identity matrix if π is the initial x-y plane of the global frame.

GNSS position measurement model

Before constructing the GNSS measurement model, the frame transformation ${}^{N}_{G}\mathbf{R}$ and ${}^{N}\mathbf{p}_G$ is required. We initialize the transformation by a simple method.

We start by the GNSS position measurement:

$${}^{\mathrm{N}}\boldsymbol{p}_{\mathrm{gnss}} = {}^{\mathrm{N}}\boldsymbol{p}_G + {}^{\mathrm{N}}_{G}\boldsymbol{R}^{\mathrm{G}}\boldsymbol{p}_{\mathrm{gnss}} \tag{24}$$

The origin of the ENU frame is set as the first position solution from GNSS sensor. We collect a sequence of VIO poses $\{{}^{G}\mathbf{p}_{I,1}, {}^{G}\mathbf{p}_{I,2}, ..., {}^{G}\mathbf{p}_{I,n}\}$ and GNSS poses $\{{}^{G}\mathbf{p}_{\mathrm{gnss},1}, {}^{G}\mathbf{p}_{\mathrm{gnss},2}, ..., {}^{G}\mathbf{p}_{\mathrm{gnss,n}}\}$ with a suitable traveling distance. By differentiating the poses, we can eliminate ${}^{N}\mathbf{p}_G$:

$${}^{N}\boldsymbol{p}_{\mathrm{gnss,n}} - {}^{\mathrm{N}}\boldsymbol{p}_{\mathrm{gnss},1} = {}^{\mathrm{N}}_{\mathrm{G}}\boldsymbol{R}({}^{\mathrm{G}}\boldsymbol{p}_{\mathrm{gnss},n} - {}^{\mathrm{G}}\boldsymbol{p}_{\mathrm{gnss},1}) \tag{25}$$

We initialize the rotation ${}^{N}_{G}\mathbf{R}$ with the yaw angle by solving Eq. (25), and then the translation ${}^{N}\mathbf{p}_G$ can also be initialized following Eq. (24). Note that we do not require a least-squares method as Lee et al. (2020) do since the simple initialization is reliable enough for ground vehicles, which can be demonstrated in the experiment section.

The subsequent position measurements are converted from the Earth-Centered Earth-Fixed (ECEF) frame to the ENU frame and construct the hybrid measurements. We loosely couple the GNSS position output in the measurement model. The measurement residual calculated in the ENU frame is given by:

$$\boldsymbol{r}_g = {}^{N}\boldsymbol{p}_{\mathrm{gnss}} - \left({}^{\mathrm{N}}_{\mathrm{G}}\boldsymbol{R}^{\mathrm{G}}\boldsymbol{p}_I + {}^{\mathrm{N}}\boldsymbol{p}_{\mathrm{G}} \right) \tag{26}$$

with the Jacobian measurement matrix:

$$\boldsymbol{H}_g = \left[\mathbf{0}_{3\times12} \ {}^{\mathrm{N}}_{\mathrm{G}}\hat{\mathbf{R}} \ \mathbf{0}_{3\times6} \left\lfloor ({}^{\mathrm{N}}_{\mathrm{G}}\hat{\mathbf{R}}^{\mathrm{G}}\hat{\mathbf{p}}_{\mathrm{I}})_{\times} \right\rfloor \ \boldsymbol{I}_3 \ \mathbf{0}_{3\times6M} \right] \tag{27}$$

In Eq. (26), the global location is directly used as the position measurement for each time step. The pose estimation will be corrected by the motion manifold constraints in the GNSS-denied environment where GNSS measurements may be inaccurate.

Adaptive filtering

For various scenarios, setting a fixed manifold constraint noise matrix in Eq. (16) is unrealistic and may worsen the pose estimation. We adopt an adaptive strategy to exploit the information on manifold constraints. It is noted from Eq. (15) that the derivative of the motion manifold is equivalent to the velocity in the body frame. Given a sequence of body velocity ${}^{b}\mathbf{v}_i, i = 1, 2, ..., n$, we suppose the real-time observation noise matrix as:

$$\hat{\mathbf{R}}_k = E\left[\boldsymbol{z}_{\mathrm{n}}\boldsymbol{z}_{\mathrm{n}}^{\mathrm{T}}\right] = \begin{bmatrix} E\left[{}^{\mathrm{b}}v_y^2\right] & 0 \\ 0 & E\left[{}^{\mathrm{b}}v_z^2\right] \end{bmatrix} \tag{28}$$

where $\mathbf{z}_n = [{}^b v_y \quad {}^b v_z]^T$. In Eq. (28) ${}^b v_y$ and ${}^b v_z$ are assumed to be independent. We use Root Mean Square (RMS) as the criterion of the algorithm to evaluate $\hat{\mathbf{R}}_k$ assuming that the expectation of ${}^b v_y$ and ${}^b v_z$ is zero:

$$\begin{cases} \overline{v}_y = \sqrt{E\left[{}^b v_y^2\right]} = \sqrt{\dfrac{\sum_{i=1}^{n} ({}^b v_{y,i})^2}{n}} \\ \overline{v}_z = \sqrt{E\left[{}^b v_z^2\right]} = \sqrt{\dfrac{\sum_{i=1}^{n} ({}^b v_{z,i})^2}{n}} \end{cases} \tag{29}$$

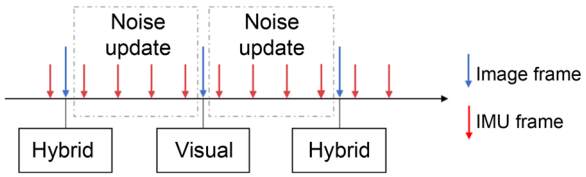

Fig. 2 Hybrid measurement update. Once comes a new image frame, the manifold noise is updated. However, the motion manifold constraint is only performed when the four conditions in Algorithm 1 are met

Therefore, we propose an adaptive filtering algorithm for the motion manifold, as shown in Algorithm. 1. For each iteration, we check four conditions from C_1 to C_4. C_1 means that the manifold constraints are not required under low speed motions, and C_2 determines whether the current state conforms to the manifold assumption in Eq. (15). C_3 is

Algorithm 1 Adaptive filtering algorithm for motion manifold constraint

Input: feature tracking rate r_f body velocity $\left\{ {}^b\boldsymbol{v}_i = \left[{}^b v_{x,i} \quad {}^b v_{y,i} \quad {}^b v_{z,i} \right]\right\}$ between current image frame and previous image frame, observation noise n_y and n_z, $i = 1, 2, \ldots, n$

Parameter: body velocity threshold v_{th}, tracking rate threshold r_{th}

1: Noise approximation: calculate $\overline{v}_y$ v_y and $\overline{v}_z$ using Eq. (29)

2: Initialize the motion manifold measurement model $\boldsymbol{H}_m$, $\boldsymbol{r}_m$

3: Check states:

$C_1: {}^b v_{x,n} \geq 1$, $C_2: {}^b v_{y,n}, {}^b v_{z,n} \leq v_{\mathrm{th}}$

$C_3: \overline{v}_y \geq \sqrt{n_y}, \overline{v}_z \geq \sqrt{n_z}$, $C_4: r_f \leq r_{\mathrm{th}}$

4: **if** $C_1 \,\&\, C_2 \,\&\, C_3 \,\&\, C_4$ **then**

5: $\boldsymbol{H}_m := \left[\boldsymbol{H}_{\mathrm{rot}}^{\mathrm{T}} \; \boldsymbol{H}_{v,e,l}^{\mathrm{T}} \; \boldsymbol{H}_{\mathrm{tran}}^{\mathrm{T}} \right]^{\mathrm{T}}$

$\boldsymbol{r}_m := \left[\boldsymbol{r}_{\mathrm{rot}}^{\mathrm{T}} \; \boldsymbol{r}_{v,e,l}^{\mathrm{T}} \; \boldsymbol{r}_{\mathrm{tran}}^{\mathrm{T}} \right]^{\mathrm{T}}$

6: **else if** `checkStatic==True` **then**

7: $\boldsymbol{H}_m := \boldsymbol{H}_z$

$\boldsymbol{r}_m := \boldsymbol{r}_z$

8: **else**

9: **return**

10: **end if**

11: Construct the measurement noise matrix $\boldsymbol{R}_m$

12: **if** chi-square_test($\boldsymbol{H}_m, \boldsymbol{r}_m, \boldsymbol{R}_m$)==True **then**

13: Motion manifold measurement update

14: **end if**

15: Noise update $n_y := \overline{v}_y^2$, $n_z := \overline{v}_z^2$

16: **return**

the most critical one. We take the RMS of ${}^{G}\mathbf{v}_b$ computed in Eq. (29) as the evaluation standard of the filtering effect. In C_4 we take the feature matching ratio r_f in the frontend as the criterion to measure the quality of visual observations. Manifold constraints are enabled when r_f is low enough, which means that visual observations may have large deviations. These four conditions guarantee both effectiveness and precision in pose estimation. Intuitively, the RMS of ${}^{G}\mathbf{v}_b$ should be reduced after the measurement update is completed. In the next stage of the hybrid measurement update (visual update and manifold constraint), the RMS is set as the new observation noise, as shown in Fig. 2. A Mahalanobis distance test (chi-square test) is employed for all the measurements to detect and eliminate potential outliers.

Observability analysis

Observability matrix

Observability reveals whether the information provided by the sensor measurements is sufficient to estimate the states without ambiguities (Huai & Huang, 2018). Since MSCKF and EKF-SLAM use the same visual measurements and linearization operation, we can perform observability analysis from the perspective of EKF-SLAM, which has the same observability property as MSCKF. Among the estimated states of the algorithm we proposed, IMU biases can prove to be observable (Kelly & Sukhatme, 2011). Considering that the measurement model in Eq. (17) only takes effect in static scenes, we ignore it in the following analysis. Lee et al. (2020) have proved that GNSS-VIO has the same four unobservable directions as VIO if estimating states in the VIO frame (i.e., the global frame). At the same time, it is fully observable if estimating states in the ENU frame. Hence we focus on the motion manifold constraints' impact on observability. For the sake of simplicity, we only analyze the three state variables of rotation ${}^{I}_{G}\mathbf{q}$, velocity ${}^{G}\mathbf{v}_I$ and translation ${}^{G}\mathbf{p}_I$. The observability matrix in the period $[m, m+n]$ is defined as:

$$\boldsymbol{\Theta} \triangleq \begin{bmatrix} \boldsymbol{H}_m \\ \boldsymbol{H}_m \boldsymbol{\Phi}_m \\ \vdots \\ \boldsymbol{H}_{m+n} \boldsymbol{\Phi}_{m+n-1} \cdots \boldsymbol{\Phi}_m \end{bmatrix} \tag{30}$$

where $\mathbf{H}_k$ is the measurement matrix at time step k, and $\boldsymbol{\Phi}_k$ is the state transition matrix from k to $k+1$. At time step k, the state vector $\mathbf{X}_I$ is defined as Eq. (31) if there are K features observed by the camera in the time interval $[m,\mathrm{m}+n]$ (Fig. 3):

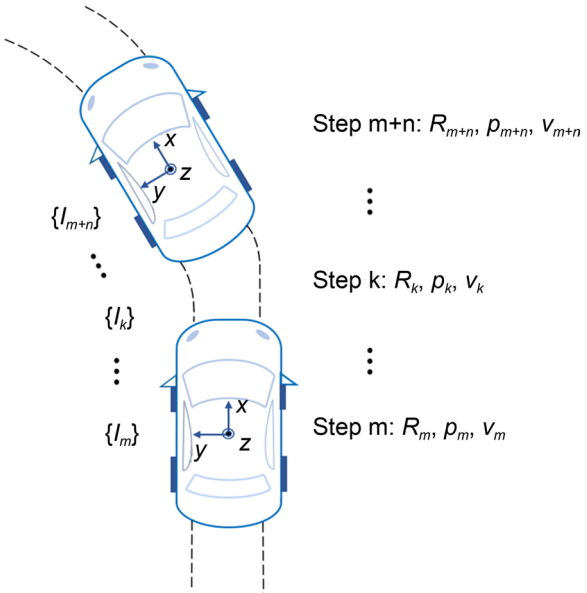

Fig. 3 State vector in the obervability analysis

$$\boldsymbol{X}_{\mathrm{I}} = \begin{bmatrix} {}^{\mathrm{I}}_{G}\boldsymbol{q}^{\mathrm{T}} & {}^{\mathrm{G}}\boldsymbol{v}_{\mathrm{I}}^{\mathrm{T}} & {}^{\mathrm{G}}\boldsymbol{p}_{\mathrm{I}}^{\mathrm{T}} & {}^{\mathrm{G}}\boldsymbol{p}_{1}^{\mathrm{T}} & \cdots & {}^{\mathrm{G}}\boldsymbol{p}_{\mathrm{K}}^{\mathrm{T}} \end{bmatrix} \tag{31}$$

The observation of the system includes visual measurements and motion manifold constraints. We denote $\mathbf{H}_v = \mathbf{H}_{vel}$ and $\mathbf{H}_p = \begin{bmatrix} \mathbf{H}_{rot}^T & \mathbf{H}_{tran}^T \end{bmatrix}^T$ for convenience. The measurement matrix is a stack of three Jacobian matrices:

$$\boldsymbol{H}_k = \begin{bmatrix} \boldsymbol{H}_{c,k}^T & \boldsymbol{H}_{v,k}^T & \boldsymbol{H}_{p,k}^T \end{bmatrix}^T \tag{32}$$

The Jacobian $\mathbf{H}_{c,k}$ contains K block rows for the form of i-th block rows:

$$\boldsymbol{H}_{c,k}^{(i)} = \begin{bmatrix} \boldsymbol{H}_{I,k}^{(i)} & \mathbf{0}_{3\times 3} & \cdots & \boldsymbol{H}_{f,k}^{(i)} & \cdots & \mathbf{0}_{3\times 3} \end{bmatrix} \tag{33}$$

while $\mathbf{H}_{v,k}$ and $\mathbf{H}_{p,k}$ are given in Eq. (34) and Eq. (35):

$$\boldsymbol{H}_{v,k} = \Lambda_1 \begin{bmatrix} \lfloor {}^{I_k}\boldsymbol{v}_{\times} \rfloor & {}^{I_k}_{G}\boldsymbol{R} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3K} \end{bmatrix} \tag{34}$$

$$\boldsymbol{H}_{p,k} = \begin{bmatrix} \Lambda_2 {}^{G}_{I_k}\boldsymbol{R} \lfloor \boldsymbol{e}_{3\times} \rfloor & \mathbf{0}_{2\times 3} & \mathbf{0}_{2\times 3} & \mathbf{0}_{2\times 3K} \\ \mathbf{0}_{1\times 3} & \mathbf{0}_{1\times 3} & \boldsymbol{e}_3^T & \mathbf{0}_{1\times 3K} \end{bmatrix} \tag{35}$$

Similarly, the block row $\boldsymbol{\Theta}_k$ of the observability matrix encompasses three parts:

Table 2 Simulation configurations

Parameter	Value
IMU frequency	100 Hz
Camera frequency	10 Hz
GNSS frequency	10 Hz
Gyroscope noise	0.0001 $rad/s/\sqrt{Hz}$
Acceleration noise	0.0005 $m/s^2/\sqrt{Hz}$
Gyroscope random walk	0.000005 $rad/s^2/\sqrt{Hz}$
Acceleration random walk	0.00004 $m/s^3/\sqrt{Hz}$
GNSS position error	0.5 m
Image width	640 pixels
Image height	640 pixels
Feature observation noise	1.5 pixels

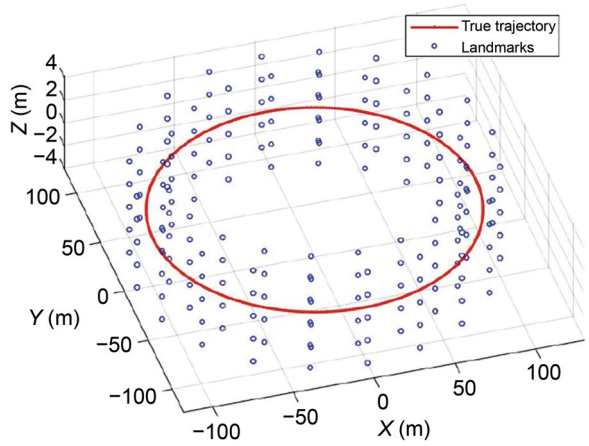

Fig. 4 True trajectory and landmarks

$$\boldsymbol{\Theta}_k = \left[\boldsymbol{\Theta}_{c,k}^{\mathrm{T}}\ \boldsymbol{\Theta}_{v,k}^{\mathrm{T}}\ \boldsymbol{\Theta}_{p,k}^{\mathrm{T}}\right]^{\mathrm{T}} \tag{36}$$

$\boldsymbol{\Theta}_{c,k}$ can refer to the conclusion in (Li & Mourikis, 2013), while $\boldsymbol{\Theta}_{v,k}$ and $\boldsymbol{\Theta}_{p,k}$ are derived in Eq. (37):

where ${}^G\mathbf{v}_g = {}^G\mathbf{v}_{I_m} + {}^G\mathbf{g}\Delta t$, $\Delta\mathbf{p} = {}^G\mathbf{p}_{I_k} - {}^G\mathbf{p}_{I_m} - {}^G\mathbf{v}_{I_m}\Delta t - \frac{1}{2}{}^G g\Delta t^2$, $\Delta t = \Delta t_{k-1} + \ldots + \Delta t_m$. According to (Li et al., 2012), the nullspace matrix of $\boldsymbol{\Theta}_{c_k}$ is given by:

$$\mathbf{N} = \begin{bmatrix} \mathbf{0}_{3\times3} & {}^{I_m}_{G}\boldsymbol{R}\,{}^G\boldsymbol{g} \\ \mathbf{0}_{3\times3} & -\left\lfloor {}^G\boldsymbol{v}_m\times\right\rfloor {}^G\boldsymbol{g} \\ \boldsymbol{I}_3 & -\left\lfloor {}^G\boldsymbol{p}_m\times\right\rfloor {}^G\boldsymbol{g} \\ \boldsymbol{I}_3 & -\left\lfloor {}^G\boldsymbol{p}_{f_1}\times\right\rfloor {}^G\boldsymbol{g} \\ \boldsymbol{I}_3 & -\left\lfloor {}^G\boldsymbol{p}_{f_2}\times\right\rfloor {}^G\boldsymbol{g} \\ \vdots & \\ \boldsymbol{I}_3 & -\left\lfloor {}^G\boldsymbol{p}_{f_K}\times\right\rfloor {}^G\boldsymbol{g} \end{bmatrix} \tag{38}$$

It can be verified that $\mathbf{N}$ is also the nullspace of $\boldsymbol{\Theta}_{v,k}$, i.e., $\boldsymbol{\Theta}_{v,k}\cdot\mathcal{N} = \mathbf{0}$, which means velocity constraints do not change the unobservable dimensions ideally. And for rotation and translation constraints, $\boldsymbol{\Theta}_{p,k}\cdot\mathcal{N} \neq \mathbf{0}$. Specifically, the z-axis of position is observable because of the plane assumption.

Observability measure

By establishing a local observability matrix in a short time, we can obtain the observability, and observable dimensions of the system (Butcher et al., 2017). Previous analysis indicates that the observable dimension of the system has not changed. The observability of the original observable dimensions can be measured by quantitative observability. Gramian matrix, which is a common measure of observability, is introduced to evaluate the observability of the EKF system with motion manifold constraints.

The Gramian matrix measures the sensitivity of the output concerning the initial condition. The discrete-time Gramian matrix is defined as:

$$\begin{cases} \boldsymbol{\Theta}_{v,k} = \boldsymbol{H}_{v,k}\boldsymbol{\Phi}_{k-1}\boldsymbol{\Phi}_{k-2}\cdots\boldsymbol{\Phi}_m = \Lambda_1\left[({}^{I_k}_{G}\boldsymbol{R}\left\lfloor({}^G\boldsymbol{v}_g)_\times\right\rfloor {}^{G}_{I_m}\boldsymbol{R}\ \ {}^{I_k}_{G}\boldsymbol{R}\ \ \mathbf{0}_{3\times(3+3K)}\right] \\ \boldsymbol{\Theta}_{c,k} = \boldsymbol{H}_{p,k}\boldsymbol{\Phi}_{k-1}\boldsymbol{\Phi}_{k-2}\cdots\boldsymbol{\Phi}_m = \begin{bmatrix} \Lambda_2({}^{G}_{I_k}\boldsymbol{R}\lfloor\boldsymbol{e}_{3\times}\rfloor {}^{I_k}_{I_m}\boldsymbol{R})\ \ \mathbf{0}_{2\times3}\ \ \mathbf{0}_{2\times3}\ \ \mathbf{0}_{2\times(3+3K)} \\ -\boldsymbol{e}_3^T\left\lfloor\Delta\boldsymbol{p}_\times\right\rfloor\ \ \boldsymbol{e}_3^T\Delta t\ \ \boldsymbol{e}_3^T\ \ \mathbf{0}_{1\times(3+3K)} \end{bmatrix} \\ \boldsymbol{\Phi}_k = \begin{bmatrix} \boldsymbol{\Phi}_{I_k} & \mathbf{0}_{9\times3K} \\ \mathbf{0}_{9\times3K} & \boldsymbol{I}_{3K\times3K} \end{bmatrix} \end{cases} \tag{37}$$

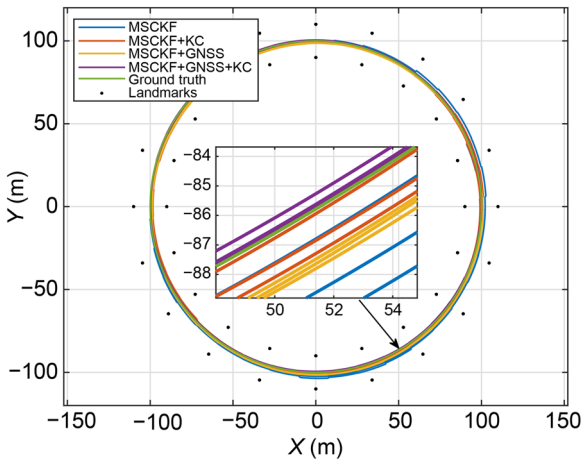

Fig. 5 Top view of the simulation trajectory

$$\begin{aligned} W_d &= \sum_{k=m}^{r} W_d(x_k,t_k) \\ &= \sum_{k=m}^{r} \boldsymbol{\Phi}^T(t_k,t_m)\boldsymbol{H}_k^T\boldsymbol{H}_k\boldsymbol{\Phi}(t_k,t_m) \end{aligned} \tag{39}$$

We take the obsevability index (OI) in Eq. (40) as the obsevability measure:

$$OI = \log_{10}(\max(\sigma(W_d))) \tag{40}$$

Motion manifold constraints can enhance observability, which will be verified in the following section.

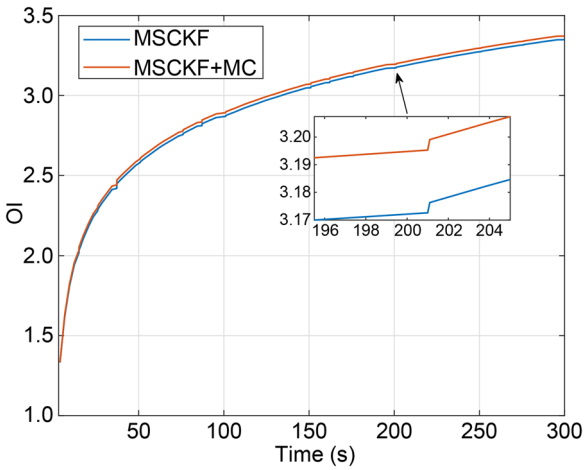

Fig. 6 Comparison of oberservability w.r.t. motion manifold constraints

Table 3 Comparison of average RMSE on different algorithms in the simulation

	Position(m)/ RR(%)	Orientation(deg)/ RR(%)
MSCKF	1.40	0.13
MSCKF+MC	1.35/3.57	0.11/15.38
MSCKF+GNSS	1.23/12.14	0.12/7.69
MSCKF+GNSS+MC	1.00/28.57	0.12/7.69

Evaluation

In order to validate the effectiveness of our proposed approach, we conduct the simulation and real world tests on different public datasets. Our experiments are conducted on a laptop with Intel(R) Core(TM) i7-10710U CPU@1.10Ghz and 16 G RAM.

Simulation

In the simulation test, we assume that the car is moving along a circle on a plane. A total of 200 landmarks are scattered around the real trajectory, as shown in Fig. 4. The car loops three times in a circle with a radius of 100 m. The landmarks are generated along the inside cylinder wall with a radius of 90 m and outside cylinder walls with a radius of 110 m. The camera captures landmarks in the field of view of the camera within a range of 20 m. We generate sensor measurements with noise according to the trajectory and landmarks. The yaw angle between the ENU and global frame is set as 10°. Specific configurations are shown in Table 2.

Position accuracy

The forward direction of the car corresponds to the y-axis of the IMU frame and the camera frame, so the body velocity in the x-axis and y-axis is constrained. In the simulation test, four algorithms are compared: standard MSCKF (benchmark), MSCKF+MC(Manifold constraints), MSCKF+GNSS, and MSCKF+GNSS+MC. The pose errors and trajectories are illustrated in Fig. 5 and Fig. 7. From Fig. 7a and b, we notice that the drift of pose error is bounded in MSCKF+GNSS and MSCKF+GNSS+MC due to the global measurements. The pose error is reduced when introducing manifold constraints, especially in the yaw and z-axis position estimation (see Fig. 7c and d), which confirms the effectiveness of the manifold constraint. Table 3 lists the total Average RMSE (ARMSE) and Reduction Rate (RR) compared to MSCKF. We can note that MSCKF+GNSS+MC achieves the highest reduction rate for the position error, which outperforms the algorithms that only combine GNSS or manifold constraints.

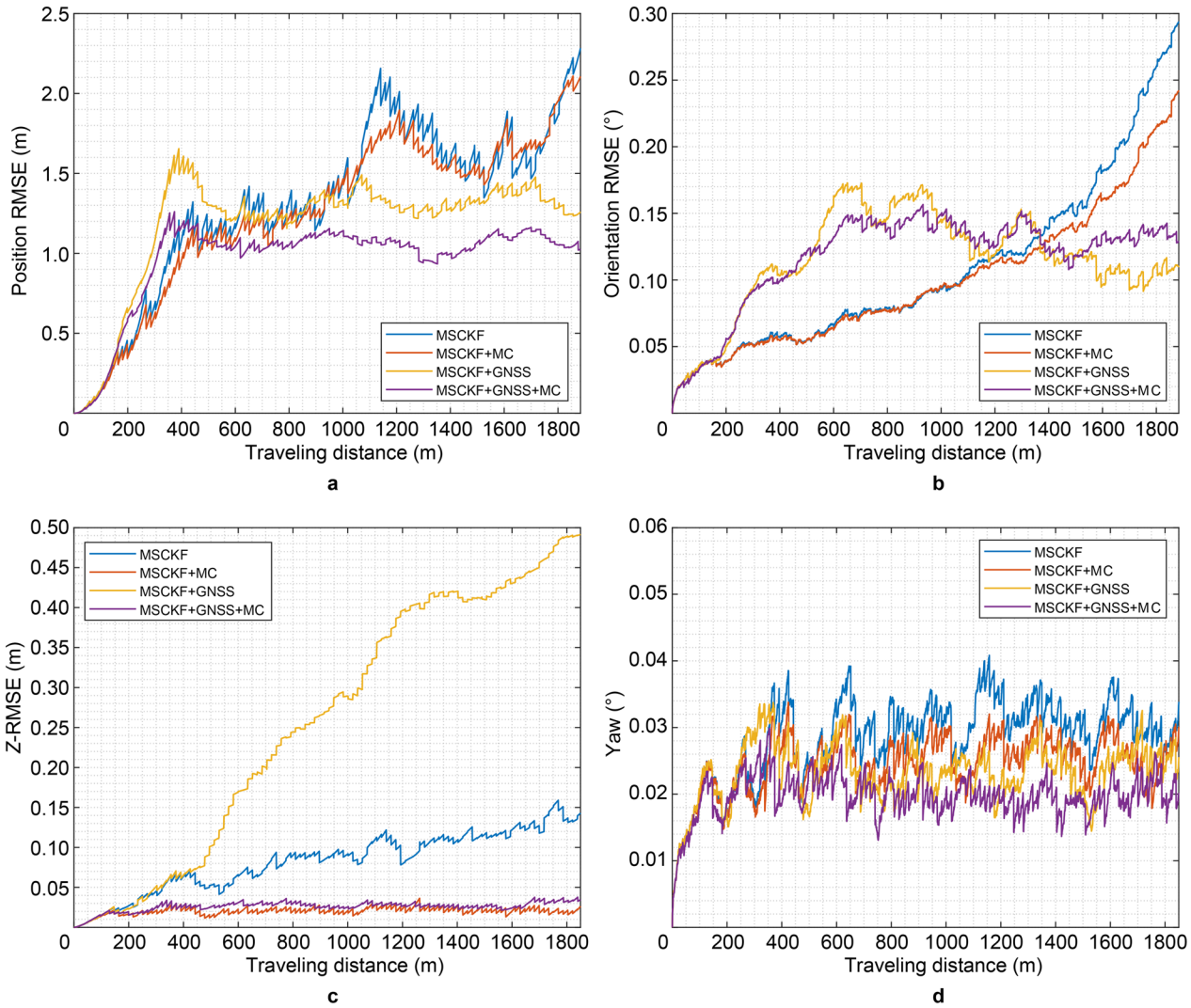

Fig. 7 Comparison of pose errors in 50 Monte Carlo runs. **a** Position ARMSE; **b** Orientation ARMSE; **c** Position ARMSE on the z-axis; **d** Yaw ARMSE

Table 4 Trajectory RMSE(m) on the Brno urban dataset

Sequence(Brno-)	Distance (m)	Alogorithm					
		Ours	Ours(w/o adaptive)	MSCKF	VINS-Mono	R-VIO	ORB_SLAM3
1_1_6_1(loops)	1480	25.37	41.07	38.75	**13.78**	30.06	32.23
1_2_1_1(turnings)	1570	**22.66**	23.02	30.24	138.80	51.70	46.09
1_2_1_2(turnings)	1570	**8.45**	15.42	51.52	24.98	57.12	×
1_2_6_1(parkings)	730	35.26	40.29	45.38	43.33	**29.78**	×
2_1_10_1(turnings)	3700	**15.04**	35.41	31.62	×[a]	82.44	×
2_1_10_3(bumps)	1150	**16.06**	19.45	32.23	185.76	29.04	×

[a] × means VIO fails to initialize or track features, and bold value means the best result

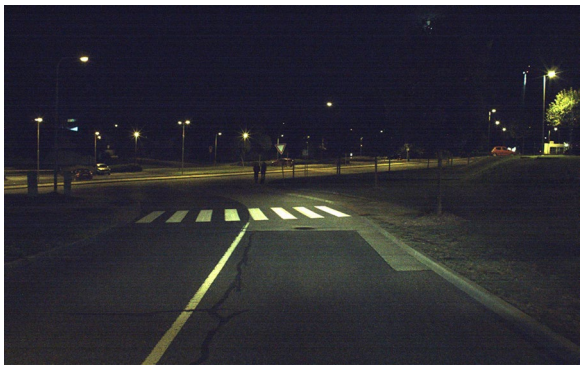

Fig. 8 A sample in Brno-2_1_10_1 (night scenario)

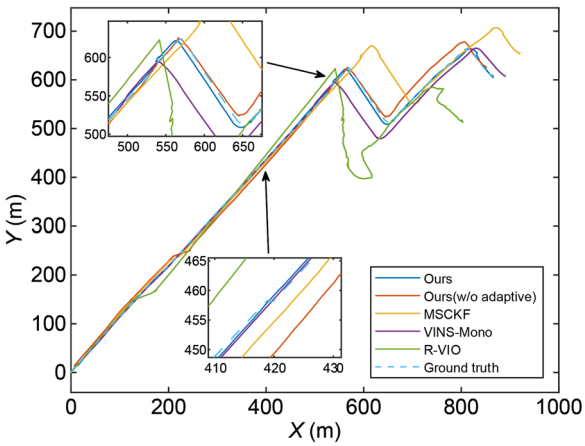

Fig. 9 Trajectories on Brno-1_2_1_2

Observability measures

We compare the observability index in Eq. (40) by calculating the Gramian matrix. We calculate the Gramian matrix for each time step, and the result is shown in Fig. 6. The observability index of MSCKF+MC is higher than that of MSCKF, indicating better observability.

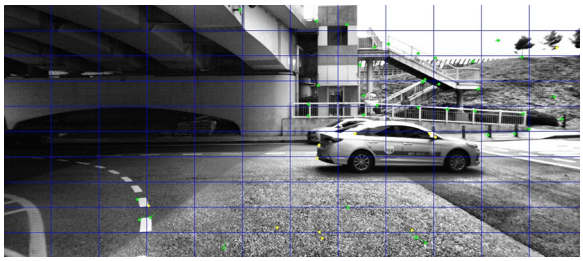

Fig. 10 A challenging section sample and the feature extraction in Kaist Urban sequence 38

Real world test

VIO+Manifold constraints

We compare the performance of motion manifold constraint-aided VIO without GNSS measurements. Both filter-based (MSCKF, R-VIO (Huai & Huang, 2018), and our algorithm without adaptive filtering strategy) and optimization-based methods (VINS-Mono, ORB_SLAM3) are included in our comparison test on Brno Urban Dataset (Ligocki et al., 2020), which provides data from four WUXGA RGB cameras with 1920×1200 pixels and 400Hz IMU. We intercept parts of datasets lasting for about three minutes, including parking, turnings, loops, bumps, and other scenes. The scenarios of sequence Brno-2_1_10_1 and sequence Brno-2_1_10_3 are at night, as shown in Fig. 8. For each trial, estimated trajectories are aligned with the ground truth trajectories using Umeyama's method (Umeyama, 1991). Table 4 concludes the errors of aligned trajectories on Brno Urban datasets, and Fig. 9 shows the estimated trajectories on a typical sequence Brno-1_2_1_2.

It can be noted that motion manifold constraints reduce the divergence of pose estimation effectively under special scenarios. In night scenarios (Brno-2_1_10_1, Brno-2_1_10_3) with scarce visual information, VINS-Mono and ORB_SLAM3 have a worse performance due to the difficulty in the convergence of visual error, while our work can make up for this disadvantage well. And in the textureless and weak light scenario (Brno-1_2_6_1), our

Table 5 Trajectory RMSE(m)/backend processing time per frame(ms) on the Kaist Urban dataset

	Kaist 30	Kaist 31	Kaist 32	Kaist 33	Kaist 38
VINS-Mono	201.44/20.02	343.69/14.18	34.72/11.29	23.07/21.29	88.00/23.90
Pure GNSS	10.53/×	7.47/×	4.85/×	9.01/×	15.26/×
GNSS-MSCKF	9.88/4.65	25.44/3.90	8.33/6.18	10.88/5.21	10.41/5.12
VINS-Fusion	39.63/27.52	23.47/18.17	6.79/14.65	5.25/27.6	10.28/27.93
Ours	**6.92**/5.01	**4.46**/4.73	**2.65**/6.59	**3.80**/5.66	**7.89**/6.34

Bold values mean the best one of RMSE for each sequence

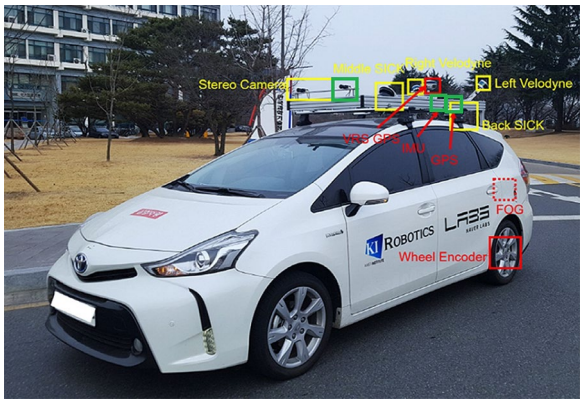

Fig. 11 Selected sensors (green boxes) in the Kaist Urban dataset Choi et al. (2018)

method also achieves a small gain with the assistance of motion manifold constraints compared with MSCKF. And in some sequences (Brno-1_1_6_1, Brno-2_1_10_1) which incorporate drastic driving motions like bumps or fast turns where fixed kinematic noise can not guarantee a positive constraint effect, whereas adopting an adaptive strategy is more reasonable. In general, our algorithm can handle different types of motion better and report more accurate results than other algorithms for most cases.

VIO+GNSS+Manifold constraints

We use Kaist Urban dataset (Jeong et al., 2019) to evaluate the performance of VIO with GNSS measurements. Kaist Urban dataset is collected with a 100Hz IMU, a 10Hz stereo camera, a commercial GPS sensor measuring the positions, and a high-precision RTK. We select the left camera, IMU, and GPS sensors for our test, as shown in Fig. 11. Originally the ground truth is generated by graph SLAM, and we transform its local coordinate into the ECEF frame by aligning it with RTK data in fixed status and further transforming the coordinate to the ENU frame. We compare our algorithm with pure GNSS, VINS-Mono, MSCKF with GNSS, and VINS-Fusion (Qin et al., 2018) on five Kaist Urban sequences, among which Kaist 31 and 38 have a long distance of 10.7 km and 11 km respectively. Note that we select the first static scene as the starting point of the trajectories because of the requirement for static initialization in MSCKF. Figure 10 shows an example of challenging sections and feature extraction in Kaist Urban dataset. For time efficiency comparison, we count the total processing time of the algorithms in the backend for each trial and calculate the average time per image frame. The trajectories and numerical results are reported in Fig. 12 and Table 5.

It is noted that the VIO pose estimation achieves a higher accuracy with GNSS measurements. Compared with VINS-Fusion which does not explicitly initialize the ENU to VIO frame transformation, our algorithms demonstrate its comparative effect on both accuracy and time efficiency. There are some cases that VINS Fusion's performance is much worse than pure GNSS (Kaist 30, Kaist 31) since the VIO pose has a large deviation dragging down the fusion accuracy with GNSS. Compared with GNSS-MSCKF, our proposed algorithm shows a moderate decrease in RMSE in Kaist 31 and Kaist 38, which demonstrates the effectiveness of the motion manifold constraints. In terms of backend processing time, optimization methods like VINS-Mono and VINS-Fusion are much more time-consuming compared with filtering methods (GNSS-MSCKF and our proposed method). Although our methods consume more time (about 1.2ms per frame) than

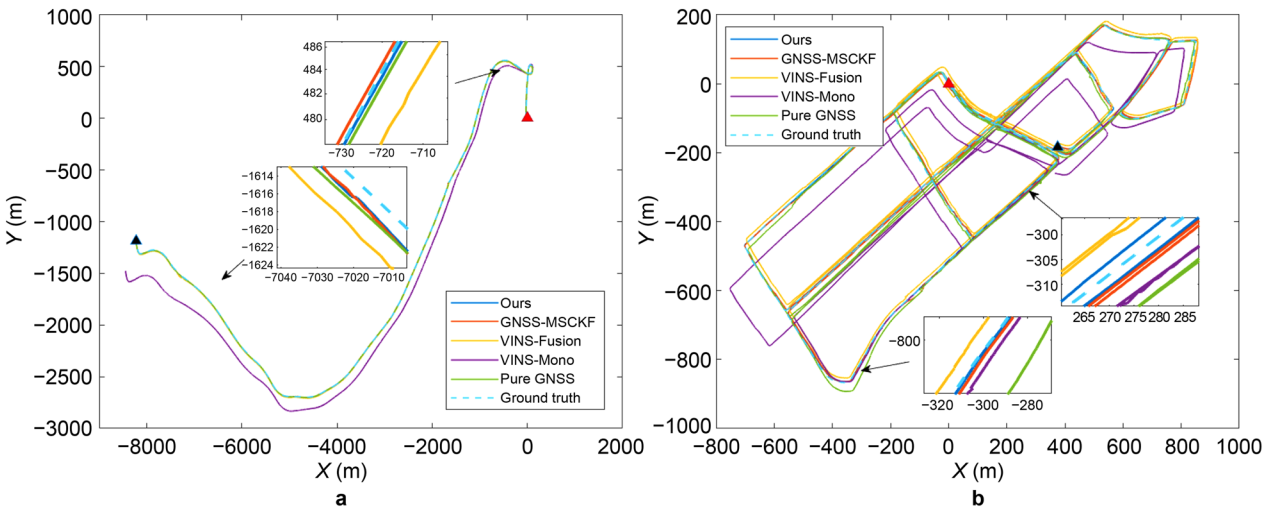

Fig. 12 Trajectories on Kaist Urban sequences with the start (red triangle) and the end (black triangle). **a** Kaist 31; **b** Kaist 38

GNSS-MSCKF, the extra time spent is acceptable. To sum up, our algorithm strikes a better balance between accuracy and time efficiency.

Conclusion

This paper presents a GNSS-aided VIO which loosely couples the GNSS position measurements. We formulate a unified framework on motion manifold to represent multiple motion manifold constraints which are specified under additional conditions. To address the challenge of time-varying noise for ground vehicles, we propose an adaptive filtering method for motion manifold constraints and derive the observability of the system, including the observability matrix for motion manifold constraints. Simulation and real world tests demonstrate the effectiveness of our system. In the future, we will introduce more robust visual estimation methods and achieve better scenario adaptability.

Acknowledgements
We express thanks to Tao Li and Jie Yin who provided the dataset, and Shanghai Key Laboratory of Navigation and Location Based Sevice, Shanghai Jiao Tong University.

Funding
This work was supported in part by the National Nature Science Foundation of China (NSFC) under Grant No. 62273229, and in part by the Equipment Pre-Research Field Foundation under Grant No. 80913010303.

Availability of data and materials
The datasets used and analysed in this study are available from the corresponding author on reasonable request.

Declarations

Competing interests
The authors declare that they have no competing interests.

Received: 18 December 2022 Accepted: 27 March 2023
Published online: 18 May 2023

References

Bloesch, M., Omari, S., Hutter, M., Siegwart, R.: Robust visual inertial odometry using a direct ekf-based approach. In: *IEEE/RSJ International Conference on Intelligent Robots & Systems*, pp. 298–304 (2015).

Butcher, E. A., Wang, J., & Lovell, T. A. (2017). On kalman filtering and observability in nonlinear sequential relative orbit estimation. *Journal of Guidance, Control, and Dynamics, 40*(9), 2167–2182.

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.

Choi, Y., Kim, N., Hwang, S., Park, K., Yoon, J. S., An, K., & Kweon, I. S. (2018). Kaist multi-spectral day/night data set for autonomous and assisted driving. *IEEE Transactions on Intelligent Transportation Systems, 19*(3), 934–948.

Cioffi, G., Scaramuzza, D. (2020). Tightly-coupled fusion of global positional measurements in optimization-based visual-inertial odometry. In: *2020 IEEE/RSJ International conference on intelligent robots and systems (IROS), IEEE*, pp. 5089–5095.

Fischler, M. A., & Bolles, R. C. (1981). Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM, 24*(6), 381–395.

Forster, C., Carlone, L., Dellaert, F., Scaramuzza, D. (2015). Imu preintegration on manifold for efficient visual-inertial maximum-a-posteriori estimation. Georgia Institute of Technology

Gong, Z., Liu, P., Wen, F., Ying, R., Ji, X., Miao, R., & Xue, W. (2020). Graph-based adaptive fusion of gnss and vio under intermittent gnss-degraded environment. *IEEE Transactions on Instrumentation and Measurement, 70*, 1–16.

Grisetti, G., Kümmerle, R., Strasdat, H., Konolige, K. (2011). g2o: A general framework for (hyper) graph optimization. In: Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), pp. 9–13.

Huai, Z., Huang, G. (2018). Robocentric visual-inertial odometry. In: *2018 IEEE/RSJ International conference on intelligent robots and systems (IROS), IEEE*, pp. 6319–6326

Jeong, J., Cho, Y., Shin, Y.-S., Roh, H., & Kim, A. (2019). Complex urban dataset with multi-level sensors from highly diverse urban environments. *The International Journal of Robotics Research, 38*(6), 642–657.

Kelly, J., & Sukhatme, G. S. (2011). Visual-inertial sensor fusion: Localization, mapping and sensor-to-sensor self-calibration. *The International Journal of Robotics Research, 30*(1), 56–79.

Lee, W., Eckenhoff, K., Geneva, P., Huang, G. (2020). Intermittent gps-aided vio: Online initialization and calibration. In: *2020 IEEE International conference on robotics and automation (ICRA), IEEE*, pp. 5724–5731.

Lee, W., Geneva, P., Yang, Y., Huang, G. (2022). Tightly-coupled gnss-aided visual-inertial localization. In: *2022 International Conference on Robotics and Automation (ICRA), IEEE*, pp. 9484–9491.

Leutenegger, S., Furgale, P., Rabaud, V., Chli, M., Siegwart, R. (2013). Keyframe-based visual-inertial slam using nonlinear optimization. In: *Proceedings of Robotics: Science and Systems*.

Li, J., Pei, L., Zou, D., Xia, S., Wu, Q., Li, T., Sun, Z., & Yu, W. (2020). Attention-slam: A visual monocular slam learning from human gaze. *IEEE Sensors Journal, 21*(5), 6408–6420.

Li, M., & Mourikis, A. I. (2013). High-precision, consistent ekf-based visual-inertial odometry. *The International Journal of Robotics Research, 32*(6), 690–711.

Li, T., Pei, L., Xiang, Y., Yu, W., & Truong, T.-K. (2022). P^3-vins: Tightly-coupled ppp/ins/visual slam based on optimization approach. *IEEE Robotics and Automation Letters, 7*(3), 7021–7027.

Li, X., Wang, X., Liao, J., Li, X., Li, S., & Lyu, H. (2021). Semi-tightly coupled integration of multi-gnss ppp and s-vins for precise positioning in gnss-challenged environments. *Satellite Navigation, 2*(1), 1–14.

Li, Y., Niu, X., Zhang, Q., Cheng, Y., & Shi, C. (2012). Observability analysis of non-holonomic constraints for land-vehicle navigation systems. In: *Proceedings of the 25th International technical meeting of the satellite division of the institute of navigation (ION GNSS 2012)*, pp. 1521–1529.

Ligocki, A., Jelinek, A., & Zalud, L. (2020). Brno urban dataset-the new data for self-driving agents and mapping tasks. In: *2020 IEEE International Conference on Robotics and Automation (ICRA), IEEE*, pp. 3284–3290.

Liu, J., Gao, W., Hu, Z. (2021). Optimization-based visual-inertial slam tightly coupled with raw gnss measurements. In: *2021 IEEE International Conference on Robotics and Automation (ICRA), IEEE*, pp. 11612–11618.

Lucas, B.D., & Kanade, T. (1997). An iterative image registration technique with an application tostereo vision. In: *Proceedings of the 7th International joint conference on artificial intelligence*.

Ma, F., Shi, J., Yang, Y., Li, J., & Dai, K. (2019). Ack-msckf: Tightly-coupled ackermann multi-state constraint kalman filter for autonomous vehicle localization. *Sensors, 19*(21), 4816.

Mourikis, A.I., Roumeliotis, S.I., et al. (2007). A multi-state constraint kalman filter for vision-aided inertial navigation. In: *ICRA*, vol. 2, p. 6.

Ning, Y., Sang, W., Yao, G., Bi, J., & Wang, S. (2021). Gnss/mimu tightly coupled integrated with improved multi-state zupt/dzupt constraints for a land vehicle in gnss-denied enviroments. *International Journal of Image and Data Fusion, 12*(3), 226–241.

Niu, X., Nassar, S., & El-Sheimy, N. (2007). An accurate land-vehicle mems imu/gps navigation system using 3d auxiliary velocity updates. *Navigation, 54*(3), 177–188.

Nüchter, A., Lingemann, K., Hertzberg, J., & Surmann, H. (2007). 6d slam-3d mapping outdoor environments. *Journal of Field Robotics, 24*(8–9), 699–722.
Panahandeh, G., Zachariah, D., Jansson, M.: Exploiting ground plane constraints for visual-inertial navigation. In: *Proceedings of the 2012 IEEE/ION Position, location and navigation symposium, IEEE*, pp. 527–534 (2012).
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.
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.
Ribeiro, M. I. (2004). Kalman and extended kalman filters: Concept, derivation and properties. *Institute for Systems and Robotics, 43*, 46.
Shin, E.-H., El-Sheimy, N. (2002). Accuracy improvement of low cost ins/gps for land applications. In: *Proceedings of the 2002 national technical meeting of the institute of navigation*, pp. 146–157.
Sukkarieh, S. (2000). Low cost, high integrity, aided inertial navigation systems for autonomous land vehicles. PhD thesis. The University of Sydney, Australia.
Sun, K., Mohta, K., Pfrommer, B., Watterson, M., Liu, S., Mulgaonkar, Y., Taylor, C. J., & Kumar, V. (2018). Robust stereo visual inertial odometry for fast autonomous flight. *IEEE Robotics and Automation Letters, 3*(2), 965–972.
Tian, R., Zhang, Y., Zhu, D., Liang, S., Coleman, S., Kerr, D. (2021). Accurate and robust scale recovery for monocular visual odometry based on plane geometry. In: *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5296–5302. https://doi.org/10.1109/ICRA48506.2021.9561215
Umeyama, S. (1991). Least-squares estimation of transformation parameters between two point patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 13*(04), 376–380.
Wan, E.A., Van Der Merwe, R. (2000). The unscented kalman filter for nonlinear estimation. In: *Proceedings of the IEEE 2000 Adaptive Systems for Signal Processing, Communications, and Control Symposium, IEEE*, (Cat. No. 00EX373), pp. 153–158.
Wu, K.J., Guo, C.X., Georgiou, G., Roumeliotis, S.I. (2017). Vins on wheels. In: *2017 IEEE International conference on robotics and automation (ICRA), IEEE*, pp. 5155–5162.
Xiaochen, Q., Zhang, H., & Wenxing, F. (2020). Lightweight hybrid visual-inertial odometry with closed-form zero velocity update. *Chinese Journal of Aeronautics, 33*(12), 3344–3359.
Xiong, L., Kang, R., Zhao, J., Zhang, P., Xu, M., Ju, R., Ye, C., & Feng, T. (2021). G-vido: A vehicle dynamics and intermittent gnss-aided visual-inertial state estimator for autonomous driving. *IEEE Transactions on Intelligent Transportation Systems, 23*(8), 11845–11861.
Yu, Z., Zhu, L., Lu, G. (2021). Vins-motion: Tightly-coupled fusion of vins and motion constraint. In: *2021 IEEE International conference on robotics and automation (ICRA), IEEE*, pp. 7672–7678.
Zhang, M., Zuo, X., Chen, Y., Liu, Y., & Li, M. (2021). Pose estimation for ground robots: On manifold representation, integration, reparameterization, and optimization. *IEEE Transactions on Robotics, 37*(4), 1081–1099.
Zhang, Z., Niu, X., Tang, H., Chen, Q., & Zhang, T. (2021). Gnss/ins/odo/wheel angle integrated navigation algorithm for an all-wheel steering robot. *Measurement Science and Technology, 32*(11), 11512.
Zuiderveld, K. (1994). Contrast Limited Adaptive Histogram Equalization. In *Graphics gems IV*, pp. 474–485.

Publisher's Note