J. Meas. Sci. Instrum., 2024, 15(1): 83-94.
DOI: 10.62756/jmsi.1674-8042.2024009
http://jmsi.publish.founderss.cn/
jmsi@nuc.edu.cn

Planar environmental constraints aided monocular visual inertial odometry

DUO Jingyun[1], ZHAO Yilin[2], ZHAO Long[2], LI Juntao[1*]

1. Beijing Key Laboratory of Intelligent Logistics System, Beijing Wuzi University, Beijing 101149, China;
2. Automation Science and Electrical Engineering, Beihang University, Beijing 100191, China

***Corresponding author:** LI Juntao (lijuntao@bwu.edu.cn)

Received: September 11, 2023 **Revised:** November 6, 2023 **Accepted:** November 23, 2023

Abstract: Motivated by the goal of enhancing the accuracy and robustness of visual inertial navigation systems (VINSs) across a wide spectrum of dynamic scenarios, protracted missions and expansive navigation ranges, we designed a monocular visual inertial odometry (VIO) augmented by planar environmental constraints. To attain efficient feature extraction and precise feature tracking, we employed a methodology that involved the extraction and tracking of uniformly distributed using features from accelerated segment test (FAST) feature points from video images, with the subsequent removal of outliers through symmetric optical flow. Additionally, we outlined the process of identifying coplanar feature points from the sparse feature set, enabling efficient plane detection and fitting. This approach constructed spatial geometric constraints on the three-dimensional coordinates of visual feature points without resorting to computationally expensive dense depth mapping. The heart of this method lied in the formulation of a comprehensive cost function, which integrated the reprojection error of visual feature points, the coordinate constraints derived from coplanar feature points, and the inertial measurement unit (IMU) pre-integration error. These integrated measurements were then utilized to estimate the system states through a nonlinear optimization methodology. To validate the accuracy and effectiveness of the proposed approach, extensive experiments were conducted using publicly available datasets and large-scale outdoor scenes. The experimental results conclusively demonstrate that compared to VINS-Mono and ORB-SLAM3, the proposed method achieves higher positioning accuracy. It can deliver precise and stable navigation results even in challenging conditions, thereby imparting significant practical value to the fields of robotics and unmanned driving.

Key words: visual inertial odometry (VIO); planar environmental constraint; state estimation; nonlinear optimization

0 Introduction

As an essential component of intelligent manufacturing, robots have garnered considerable interest and found widespread applications[1-3]. Precise navigation of robots, as a key robotics technology, has become a research hotspot in recent years. In traditional robot navigation systems, global navigation satellite system (GNSS) is commonly used for calibrating inertial navigation system (INS) to obtain drift-free state estimations. However, in practical applications, GNSS signals are susceptible to interference or deception. When GNSS signals are lost or interrupted, the state estimations provided only by INS will rapidly drift. Therefore, driven by the requirements for accurate and robust positioning, many researchers are focusing on exploring alternative navigation methods[4-6] that can provide continuous and reliable positioning, navigation and timing (PNT) information in environments with degraded GNSS signals.

In recent years, rapid advancements in computer vision technology and microprocessor computing power have elevated the visual navigation system[7-9] to a prominent research area, particularly in environments with degraded GNSS signals. The camera, as a passive sensor, offers two significant advantages over other sensors. Firstly, its passive nature renders it less susceptible to interference, thereby conferring significant value in military applications. Secondly, cameras are cost-effective, compact, and capable of capturing abundant information, enabling effective environment perception. Furthermore, the inertial measurement unit (IMU) provides state estimations by integrating acceleration and angular velocity. While it may accumulate errors over extended periods, it excels in providing highly precise short-term relative displacement. The combination of camera and IMU leverages their complementary strengths, resulting in a visual inertial integrated navigation system that offers superior positioning accuracy and reliability compared to single navigation

systems. Therefore, over the last two decades, significant progress has been made in visual inertial odometry (VIO) and visual inertial simultaneous localization and mapping (VISLAM), resulting in the development of numerous advanced systems[10-15]. While both VIO and VISLAM have their merits, VISLAM offers additional features, such as global 3D map generation and maintenance, which enhance positioning accuracy and 3D reconstruction performance. However, it comes with the trade-off of requiring higher memory and computation costs. As a consequence, VIO has gained popularity in resource-constrained platforms due to its efficiency.

Visual inertial integrated navigation methods can be broadly categorized into two groups: optimization-based methods[10-12] and filter-based methods[13-15]. Among the filter-based methods, the multi-state constraint Kalman filter (MSCKF) [13] stands out as the most popular choice. In MSCKF, the state comprises multiple poses within a sliding window, and observations are derived from the reprojection errors of visual feature points. State constraints are then established based on these feature points' observations across multiple frames. To manage computational costs, MSCKF updates the states only once, resulting in a significant error during the linearization process. In contrast, optimization-based methods estimate the states through multiple iterations. These methods incorporate both sensor poses and positions of feature points observed by cameras into their state estimation. To manage the trade-off between complexity and accuracy, a sliding window is constructed to limit the number of current states, and the historical states are converted to prior information through marginalization. As a result, optimization-based methods tend to be more accurate than filter-based methods, especially when sufficient computational power is available to support their execution.

The state-of-the-art visual inertial navigation methods primarily rely on constructing associations between adjacent frames based on feature points. While robust feature points detection and tracking can enhance positioning accuracy and stability, these methods face challenges in adapting to wide dynamic, long-endurance and large-range navigation tasks[16]. The difficulty arises from the susceptibility of visual observations collected by the camera to complex application scenarios, such as illumination changes, weak texture and dynamic targets. Additionally, longer trajectories introduce more error accumulations to the systems, further complicating their performance in such scenarios. To enhance the accuracy and stability of positioning in complex environments, wheel odometer measurements are integrated into the VIO system to calibrate accelerometer biases. For example, the incorporation of wheel odometer measurement notably improved navigation performance, especially in systems equipped with low-cost IMUs[17,18]. However, it is important to note that this approach is primarily suitable for ground vehicles, limiting its applicability to other types of vehicles or platforms. Maity et al. presented edge SLAM[19], where they introduced edge feature points instead of generalized feature points. Similarly, He et al. proposed point-line based (PL)-VIO[20] by incorporating feature lines alongside feature points. These innovative approaches enriched the feature set, leading to enhanced accuracy and robustness of positioning, particularly in weakly textured environments. However, one limitation in the mentioned works is that the introduced features are mostly treated independently, disregarding the potential structural relationships between different features.

As a higher-level spatial geometry constraint, planar constraints can be utilized to restrict the movement or placement of feature points within a three-dimensional space. Since a single monocular camera lacks the capability to directly determine pixel depth, depth sensors such as RGB-D cameras[21,22] and LiDAR[23,24] offer significant advantages in detecting the planar environment. In recent years, there has been a growing focus on training deep learning networks[25,26] to detect planes, and this approach has been successfully integrated into VIO systems. For example, robust plane-based (RP)-VIO[27] employs a plane segmentation network to distinguish static planes in dynamic environments. However, it is essential to note that this method requires additional computational resources and its generalizability may not be entirely certain. Nevertheless, the use of deep learning networks for plane detection shows promising potential in enhancing VIO systems' performance. For a single monocular camera, Li et al. proposed point-line-plane based (PLP)-VIO[28], which leverages point features and line features as well as plane regularities to enhance the VIO system's performance. This method incorporates point-to-line, point-to-plane and line-to-plane regularities into the system. However, it is important to note that this approach necessitates adding linear and planar parameters to the state variables. As the number of constraints grows, the computation cost increases significantly, posing a potential limitation in certain scenarios.

This paper presents a monocular VIO that leverages planar environmental constraints to achieve accurate and stable performance in wide dynamic, long endurance and large range navigation tasks. The primary motivation of this

research is to explore the spatial geometric characteristics of feature points, and detect planes to constrain the spatial coordinates of feature points. The main contributions of this work can be summarized as follows:

1) We extract crucial spatial geometric characteristics from the sparse feature points, and provide a plane detection and fitting method without the need for computationally expensive dense depth maps.

2) We present a monocular visual inertial state estimation method based on a nonlinear optimization framework. The core of our method lies in integrating planar constraints into the objective function and dynamically adjusting error weights associated with visual observations. This approach results in significantly improved accuracy and adaptability in state estimation.

3) We validate the proposed VIO on both publicly available datasets and in large-scale outdoor scenes. Experimental results demonstrate that additional planar constraints can ensure consistency in the estimates, reducing drift and enhancing the overall accuracy.

1 Methods

The proposed VIO is based on a nonlinear optimization framework, and the main structure is shown in Fig. 1. The system comprises two parallel modules: the front end (red part in Fig. 1) and the back end (blue part in Fig. 1). The front end handles sensor signal preprocessing, feature point detection and tracking, while the back end focuses on plane detection and fitting as well as state estimation.

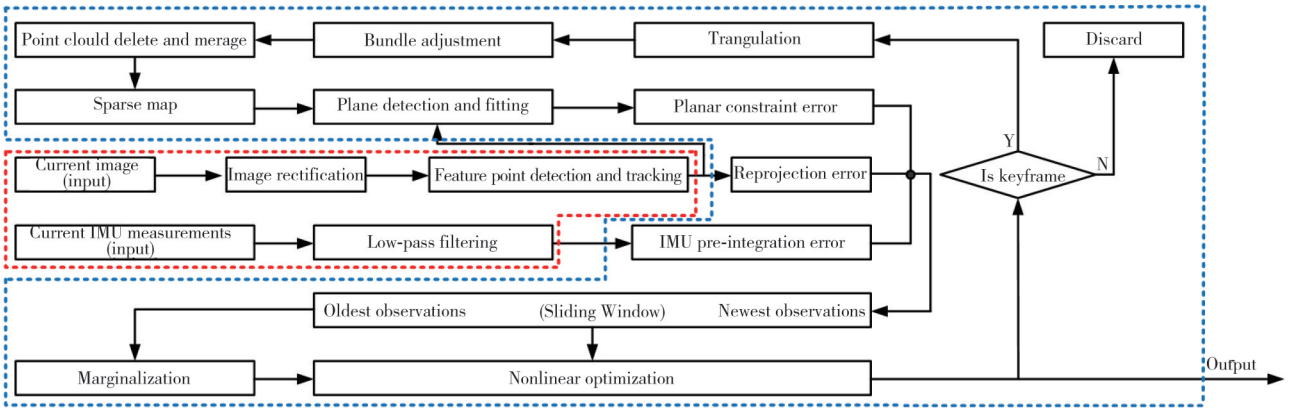


Fig. 1 Framework of proposed VIO

1.1 Feature point detection and tracking

Constructing the corresponding relations of the same-name feature points between adjacent frames is one of the key steps in VIO. This section presents the method of feature point detection and tracking used in the study.

1.1.1 Feature point detection

To enhance feature point detection, we created a 6-layer image pyramid by downsampling the original image with a scale factor of 1.2. Feature points using features from acceletate segment test (FAST) [29] are then extracted from each pyramid layer. Subsequently, we project the detected feature points back onto the original image. To ensure a uniform distribution of feature points, we partition the original image into non-overlapping grids of 40×40 (unit: pixel). The feature point with the highest intensity in each grid is selected and added to the feature point set. The results are depicted in Fig.2.

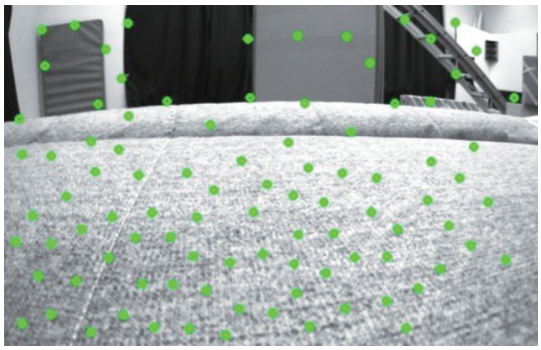


Fig. 2 Results of uniformly distributed FAST feature point detection

1.1.2 Feature point tracking

Due to small image distortions in adjacent frames, we utilize Lucas-Kanade (LK) optical flow to track feature points. Instead of using the random sample consensus (RANSAC) method employed in robust and versatile monocular visual-inertial state estimator (VINS-Mono) [11], this study adopts symmetrical optical flow to eliminate outliers, and the results are shown in Fig. 3. We calculate the optical flow of feature points at time t_k with respect to time t_{k+1}, as well as the optical flow at time t_{k+1} with respect to time t_k. The computed results

must satisfy the condition of equal magnitudes and opposite directions. If this condition is not met, the corresponding feature points can be considered as outliers and subsequently removed.

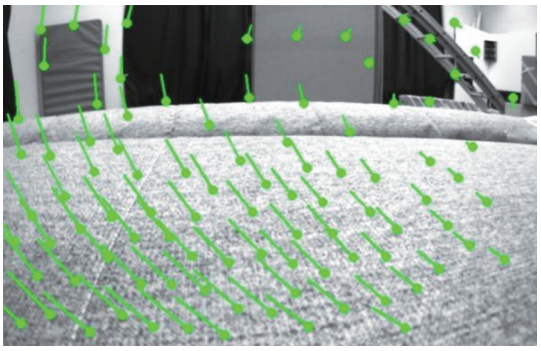


Fig. 3 Results of feature point tracking with outliers eliminated using symmetrical optical flow

To enhance the computational efficiency of symmetric optical flow, we utilize the inverse optical flow[30] when calculating the optical flow at time t_{k+1} with respect to time t_k. The optimization objective function is defined as

$$error=\sum_{x}\left[I_k(\boldsymbol{W}(\boldsymbol{x};0))-I_{k+1}(\boldsymbol{W}(\boldsymbol{x};\boldsymbol{p}))\right]^2, \quad (1)$$

where $\boldsymbol{x}$ denotes a pixel with coordinates (x, y); $\boldsymbol{p}=(p_x,p_y)$ represents the optical flow vector; $\boldsymbol{W}(\boldsymbol{x};\boldsymbol{p})=(x+p_x,y+p_y)$; $I_k(\cdot)$ and $I_{k+1}(\cdot)$ represent the gray value of the image at time t_k and t_{k+1}, respectively. Since t_{k+1} is the reference time, the corresponding value of $\boldsymbol{W}(\boldsymbol{x};\boldsymbol{p})$ is a known quantity. We utilize the Gauss-Newton method to solve Eq. (1), and the Jacobian matrix can be calculated as

$$J(\boldsymbol{p})=\sum_{x}\nabla I_k\frac{\partial \boldsymbol{W}}{\partial \boldsymbol{p}}, \quad (2)$$

where ∇I_k denotes the gradient of the image at time t_k. According to Eq. (1), t_k denotes the time before the move, so its corresponding $\boldsymbol{p}$ is zero vector 0. The iteration vector in optimization can be calculated as

$$\Delta\boldsymbol{p}=-[J(\boldsymbol{p})^TJ(\boldsymbol{p})]^{-1}J(\boldsymbol{p})^T[I_k(\boldsymbol{x})-I_{k+1}(\boldsymbol{W}(\boldsymbol{x};\boldsymbol{p}))]. \quad (3)$$

Compared to the forward optical flow that requires multiple calculations of the Jacobian matrix for iterative solutions, the Jacobian matrix of the inverse optical flow is independent of vector $\boldsymbol{p}$. As a result, only the initial Jacobian matrix needs to be calculated, and it can be reused throughout the subsequent optimization process. This significantly reduces the computational burden of the optimization process.

1.2 Plane detection and fitting

We detect coplanar feature points in a sparse feature point set and fit planes to construct the high-level planar constraints, which provides a feasible scheme to improve the positioning accuracy of our monocular visual inertial integrated navigation system.

The feature point set $\boldsymbol{E}$ in the current frame consists of two distinct subsets: the history feature point set $\boldsymbol{E}_1$ (green points in Fig. 4) and the newly detected feature point set $\boldsymbol{E}_2$ (red points in Fig. 4). $\boldsymbol{E}_1$ comprises the feature points tracked from the previous frame, while $\boldsymbol{E}_2$ comprises the feature points newly detected in the current frame. Since three non-collinear points in three-dimensional space determine a plane, as seen in Fig. 4, the current frame can be segmented into several non-overlapping triangular regions by using Delaunay triangulation, and the vertices of the triangles are the feature points.

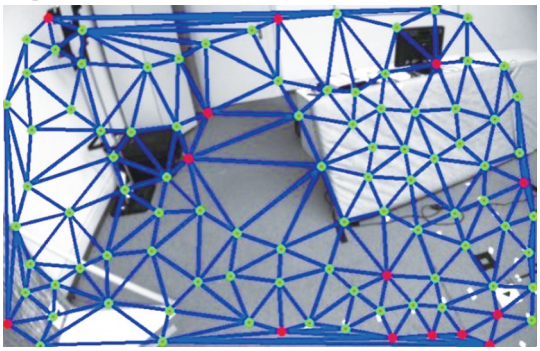


Fig. 4 Results of Delaunay triangulation. : green points representing history feature points, and red points representing newly detected feature points

Suppose the three vertices of a triangle are the feature points f_1, f_2 and f_3, respectively, and their corresponding coordinates in the world coordinate system are $\boldsymbol{p}_{f_1}^w(x_{f_1},y_{f_1},z_{f_1})$, $\boldsymbol{p}_{f_2}^w(x_{f_2},y_{f_2},z_{f_2})$ and $\boldsymbol{p}_{f_3}^w(x_{f_3},y_{f_3},z_{f_3})$, respectively. f_1, $f_2$and$f_3$can determine a plane in three-dimensional space, and the normal vector of this plane can be expressed as

$$\boldsymbol{n}=\boldsymbol{v}_1\otimes\boldsymbol{v}_2, \quad (4)$$

where $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$ denote two direction vectors on the plane, $\boldsymbol{v}_1=\boldsymbol{p}_{f_2}^w-\boldsymbol{p}_{f_1}^w$ and $\boldsymbol{v}_2=\boldsymbol{p}_{f_3}^w-\boldsymbol{p}_{f_1}^w$; and $\otimes$ denotes vector cross product. The parameters of the plane determined by f_1, f_2 and f_3 satisfy

$$\boldsymbol{N}=(a,b,c,d)^T=(\boldsymbol{n},d)^{\mathrm{T}}. \quad (5)$$

where the parameter d can be calculated by

$$d=-ax_{f_1}-by_{f_1}-cz_{f_1}. \quad (6)$$

Then vector $\boldsymbol{N}$ is normalized as

$$\bar{\boldsymbol{N}}=\frac{\boldsymbol{N}}{\|\boldsymbol{N}\|}. \quad (7)$$

The planes in the image can be detected by using the region growing algorithm. The steps are as follows:

1) Select any triangle as the initial plane seed and calculate the unit vector $\bar{\boldsymbol{N}}_1=(\bar{\boldsymbol{n}}_1,\bar{d}_1)^{\mathrm{T}}$ of the plane parameters.

2) Calculate the unit vector $\bar{\boldsymbol{N}}_2=(\bar{\boldsymbol{n}}_2,\bar{d}_2)^{\mathrm{T}}$ through adjacent triangle of the initial seed, and compare $\bar{\boldsymbol{n}}_1$ with $\bar{\boldsymbol{n}}_2$. If the angle between $\bar{\boldsymbol{n}}_1$ and $\bar{\boldsymbol{n}}_2$ is within 5 degrees, the difference between the parameters $\bar{d}_1$ and $\bar{d}_2$ continues to be compared. If the difference between $\bar{d}_1$ and $\bar{d}_2$ is less than 5%, the current triangle is merged into the initial seed. Continuously iterate over the neighboring triangles of the initial seed and repeat the previous process until no new triangles can be merged. A plane is considered to be detected if the number of triangles in the plane exceeds the threshold of 8.

3) Take a triangle that does not satisfy the coplanar condition as a new plane seed and repeat step 2) until all triangles in the image have been traversed.

1.3 State estimation

Our method tightly fuses visual data with inertial measurements and effectively estimates the states using a nonlinear optimization method. As illustrated in Fig.5, we construct the objective function by integrating reprojection errors of feature points, position constraints of coplanar feature points, and IMU pre-integration errors. The objective function of our monocular visual inertial navigation system can be expressed as

$$J(\chi)=\sum_{k=1}^{n}\sum_{i=1}^{m}\left\|\boldsymbol{r}(\hat{\boldsymbol{z}}_{f_i}^{c_k},\chi)\right\|^2+\sum_{i=1}^{m}\left\|\boldsymbol{r}(\hat{\boldsymbol{z}}_{f_i}^{w},\chi)\right\|^2+\sum_{k=1}^{n-1}\left\|\boldsymbol{r}(\hat{\boldsymbol{z}}_{b_{k+1}}^{b_k},\chi)\right\|^2+J_{\mathrm{mar}}(\chi),\tag{8}$$

where $\chi=[\boldsymbol{x}_{b_1},\cdots,\boldsymbol{x}_{b_n},\boldsymbol{p}_{f_1}^{w},\cdots,\boldsymbol{p}_{f_m}^{w}]$ denotes the set of states in the sliding window; n denotes the number of frames in the sliding window; m denotes the number of feature points that can be observed in the sliding window; $\boldsymbol{x}_{b_k}=[\boldsymbol{p}_{b_k}^{w},\boldsymbol{v}_{b_k}^{w},\boldsymbol{R}_{wb_k},\boldsymbol{b}_{a\,b_k},\boldsymbol{b}_{g\,b_k}]$ denotes the states corresponding to the kth frame in the sliding window; $\boldsymbol{p}_{b_k}^{w}$, $\boldsymbol{v}_{b_k}^{w}$ and $\boldsymbol{R}_{wb_k}$ denote the position, velocity and rotation matrix of IMU coordinate system with respect to the world coordinate system, respectively; $\boldsymbol{b}_{a\,b_k}$ and $\boldsymbol{b}_{g\,b_k}$ denote the accelerometer and gyroscope biases, respectively; $\boldsymbol{p}_{f_i}^{w}$ denotes the position of the feature point f_i in the world coordinate system; $\boldsymbol{r}(\hat{\boldsymbol{z}}_{f_i}^{c_k},\chi)$ denotes the reprojection error of feature point f_i on the kth frame in the sliding window; $\boldsymbol{r}(\hat{\boldsymbol{z}}_{f_i}^{w},\chi)$ denotes the planar constraint error of feature point f_i; $\boldsymbol{r}(\hat{\boldsymbol{z}}_{b_{k+1}}^{b_k},\chi)$ denotes the IMU pre-integration error within the kth and $(k+1)$th frame interval in the sliding window; $J_{mar}(\chi)$ denotes the prior information generated by marginalization.

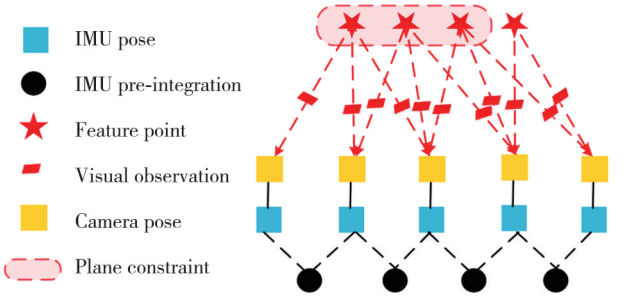


Fig. 5 Nonlinear optimization model of proposed monocular visual inertial navigation system

1.3.1 Feature point reprojection error

The reprojection error of feature point f_i on the kth frame in the sliding window can be expressed as

$$\boldsymbol{r}(\hat{\boldsymbol{z}}_{f_i}^{c_k},\chi)=\boldsymbol{C}(\boldsymbol{\pi}(\boldsymbol{R}_{bc}^{-1}(\boldsymbol{R}_{wb_k}^{-1}(\boldsymbol{p}_{f_i}^{w}-\boldsymbol{p}_{b_k}^{w})-\boldsymbol{p}_{c}^{b}))-\hat{\boldsymbol{z}}_{f_i}^{c_k}),\tag{9}$$

where $\hat{\boldsymbol{z}}_{f_i}^{c_k}$ denotes the observation of feature point f_i in the kth frame pixel coordinate system, which can be obtained by feature point tracking; $\boldsymbol{R}_{bc}$ and $\boldsymbol{p}_{c}^{b}$ are the rotation matrix and translation from the camera coordinate system to the IMU coordinate system, respectively. Denote $\boldsymbol{p}_{f_i}^{c_k}=\boldsymbol{R}_{bc}^{-1}(\boldsymbol{R}_{wb_k}^{-1}(\boldsymbol{p}_{f_i}^{w}-\boldsymbol{p}_{b_k}^{w})-\boldsymbol{p}_{c}^{b})$ as the coordinates of feature point f_i in the kth frame camera coordinate system; $\boldsymbol{\pi}(\boldsymbol{p}_{f_i}^{c_k})$ denotes the function that transforms $\boldsymbol{p}_{f_i}^{c_k}$ from the camera coordinate system to the pixel coordinate system through the internal parameter matrix; $\boldsymbol{C}$ denotes the reprojection error coefficient matrix.

In this study, we create a 6-layer image pyramid by downsampling the original image with a scale factor of 1.2. As a result, the margin of error for projecting pixels from different layers onto the original image is different. Therefore, different feature points exhibit distinct reprojection error coefficient matrices, primarily dependent on the layer number at which the feature points are detected. The reprojection error coefficient matrix $\boldsymbol{C}$ is defined as

$$\boldsymbol{C}=\begin{bmatrix}\dfrac{1}{1.2^k} & 0\\ 0 & \dfrac{1}{1.2^k}\end{bmatrix}_{2\times 2},\tag{10}$$

where k is the layer number at which the feature point f_i is detected. By introducing the projection error coefficient matrix, the weight of reprojection errors for different feature points can be adaptively adjusted in the objective function.

1.3.2 Planar constraint error

If the feature point f_i is on a spatial plane detected by Section 1.2, we substitute the coordinates of f_i into the plane equation $ax+by+cz+d=0$, and denote the

planar constraint error of feature point f_i as

$$\boldsymbol{r}(\hat{\boldsymbol{z}}_{f_i}^{w},\boldsymbol{\chi})=d-ax_{f_i}-by_{f_i}-cz_{f_i}, \tag{11}$$

where $f_i(x_{f_i},y_{f_i},z_{f_i})$ denotes the position of the feature point f_i in the world coordinate system. Note that if feature point f_i satisfies the planar constraint, we add the corresponding error to the objective function; otherwise, we simply ignore it.

1.3.3 IMU pre-integration error

Nonlinear optimization-based state estimation requires multiple iterations to update the state variables. Commonly, IMU estimates the state variables by using recursive methods. To avoid redundant IMU integration caused by changes of the initial state during optimization, we adopt the IMU pre-integration strategy[31]. It introduces relative motion increment equations that are independent of the initial state variables. The error terms of IMU pre-integration between the kth and $(k+1)$th frames in the sliding window can be represented as

$$\boldsymbol{r}(\hat{\boldsymbol{z}}_{b_{k+1}}^{b_k},\boldsymbol{\chi})=\begin{bmatrix}\boldsymbol{R}_{wb_k}^{-1}\left(\boldsymbol{p}_{b_{k+1}}^{w}-\boldsymbol{p}_{b_k}^{w}-\boldsymbol{v}_{b_k}^{w}\Delta t_{b_{k+1}b_k}+\frac{1}{2}\boldsymbol{g}^{w}\Delta t_{b_{k+1}b_k}^{2}\right)-\boldsymbol{\Delta}\hat{\boldsymbol{p}}_{b_kb_{k+1}}\\ \boldsymbol{R}_{wb_k}^{-1}\left(\boldsymbol{v}_{b_{k+1}}^{w}-\boldsymbol{v}_{b_k}^{w}+\boldsymbol{g}^{w}\Delta t_{b_{k+1}b_k}\right)-\boldsymbol{\Delta}\hat{\boldsymbol{v}}_{b_kb_{k+1}}\\ 2\left[(\boldsymbol{\Delta}\hat{\boldsymbol{q}}_{b_kb_{k+1}})^{-1}\otimes\boldsymbol{q}_{wb_k}^{-1}\otimes\boldsymbol{q}_{wb_{k+1}}\right]_{xyz}\\ \boldsymbol{b}_{a\,b_{k+1}}-\boldsymbol{b}_{a\,b_k}\\ \boldsymbol{b}_{g\,b_{k+1}}-\boldsymbol{b}_{g\,b_k}\end{bmatrix}, \tag{12}$$

where $\boldsymbol{g}^{w}$ denotes the projection of gravitational acceleration in the world coordinate system; $\Delta t_{b_{k+1}b_k}$ denotes the time interval of the kth and $(k+1)$th frames in the sliding window; $\boldsymbol{\Delta}\hat{\boldsymbol{p}}_{b_kb_{k+1}}$, $\boldsymbol{\Delta}\hat{\boldsymbol{v}}_{b_kb_{k+1}}$ and $\boldsymbol{\Delta}\hat{\boldsymbol{q}}_{b_kb_{k+1}}$ denote the pre-integral estimates of position, velocity and attitude, respectively; $[\bullet]_{xyz}$ denotes the imaginary part of a unit quaternion.

1.3.4 Sliding window and marginalization

As time progresses, the number of state variables in the visual inertial navigation system will significantly increase, leading to a considerable increase in algorithm complexity. If all state variables are solved, real-time requirements may not be met. On the other hand, solving only the state variables corresponding to the latest frame will overlook the correlations with past state variables, thus reducing the system's accuracy. In this paper, a sliding window is constructed to limit the number of state variables to be solved. The marginalization strategy is adopted to transform the remaining state variables into prior information for the state variables in the sliding window, achieving a balance between algorithm complexity and accuracy.

Suppose $\boldsymbol{\chi}_\mu$ denotes the state variables in $\boldsymbol{\chi}$ to be marginalized; $\boldsymbol{\chi}_\lambda$ denotes the state variables in $\boldsymbol{\chi}$ correlated with $\boldsymbol{\chi}_\mu$; $\boldsymbol{\chi}_p$ denotes the remaining state variables in $\boldsymbol{\chi}$ after excluding $\boldsymbol{\chi}_\mu$ and $\boldsymbol{\chi}_\lambda$. In other words, $\boldsymbol{\chi}$ is composed of $\boldsymbol{\chi}_\mu$, $\boldsymbol{\chi}_\lambda$ and $\boldsymbol{\chi}_p$, expressed as $\boldsymbol{\chi}=[\boldsymbol{\chi}_\mu,\boldsymbol{\chi}_\lambda,\boldsymbol{\chi}_p]$. As $\boldsymbol{\chi}_p$ is independent of $\boldsymbol{\chi}_\mu$, the increment equations of the Gauss-Newton method can be simplified to

$$\begin{bmatrix}\boldsymbol{H}_{\mu\mu} & \boldsymbol{H}_{\mu\lambda}\\ \boldsymbol{H}_{\lambda\mu} & \boldsymbol{H}_{\lambda\lambda}\end{bmatrix}\begin{bmatrix}\delta\boldsymbol{\chi}_\mu\\ \delta\boldsymbol{\chi}_\lambda\end{bmatrix}=\begin{bmatrix}\boldsymbol{b}_\mu\\ \boldsymbol{b}_\lambda\end{bmatrix}. \tag{13}$$

By applying the Schur complement principle[32], Eq. (13) can be simplified to

$$\begin{bmatrix}\boldsymbol{H}_{\mu\mu} & \boldsymbol{H}_{\mu\lambda}\\ 0 & \boldsymbol{H}_{\lambda\lambda}^{*}\end{bmatrix}\begin{bmatrix}\delta\boldsymbol{\chi}_\mu\\ \delta\boldsymbol{\chi}_\lambda\end{bmatrix}=\begin{bmatrix}\boldsymbol{b}_\mu\\ \boldsymbol{b}_\lambda^{*}\end{bmatrix}, \tag{14}$$

where $\boldsymbol{H}_{\lambda\lambda}^{*}=\boldsymbol{H}_{\lambda\lambda}-\boldsymbol{H}_{\lambda\mu}\boldsymbol{H}_{\mu\mu}\boldsymbol{H}_{\mu\lambda}$, and $\boldsymbol{b}_\lambda^{*}=\boldsymbol{b}_\lambda-\boldsymbol{H}_{\lambda\mu}\boldsymbol{H}_{\mu\mu}^{-1}\boldsymbol{b}_\mu$. Solving the equation $\boldsymbol{H}_{\lambda\lambda}^{*}\delta\boldsymbol{\chi}_\lambda=\boldsymbol{b}_\lambda^{*}$, the state increments $\delta\boldsymbol{\chi}_\lambda$ can be obtained to update the state variable $\boldsymbol{\chi}_\lambda$.

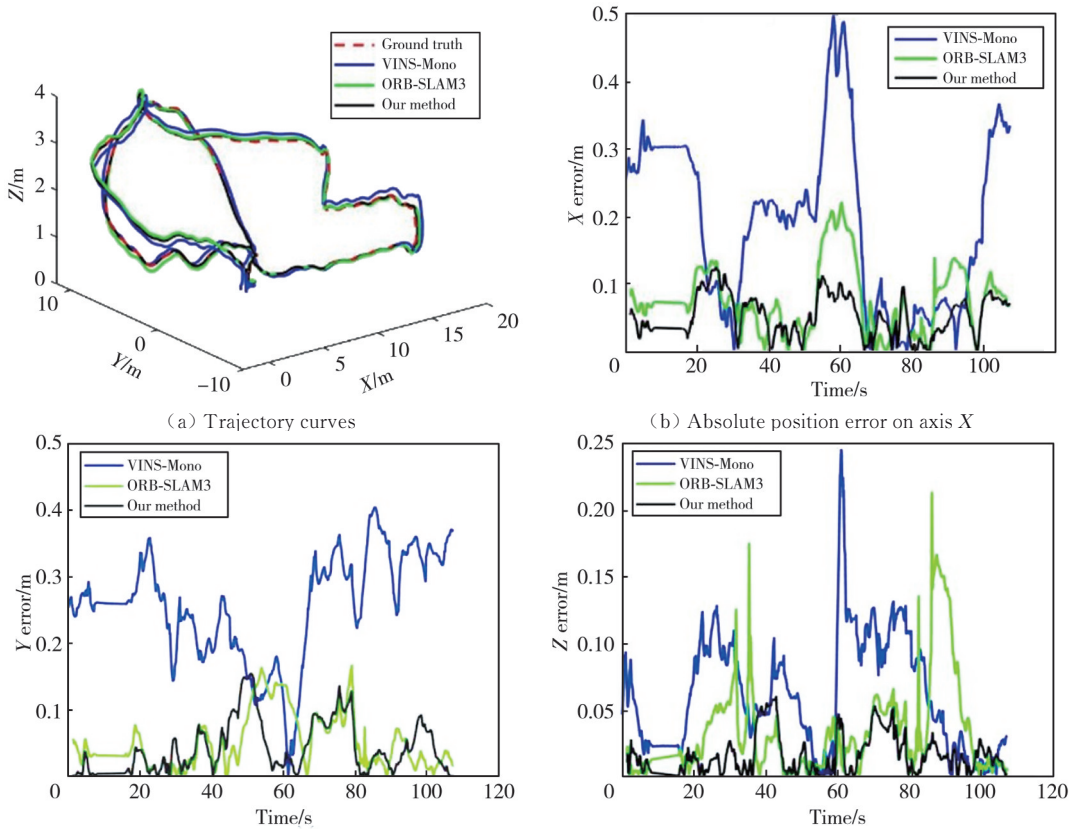
2 Experimental results

In this section, we conducted experiments to evaluate the accuracy and effectiveness of the proposed VIO. The experiments were performed on both publicly available datasets and in large-scale outdoor scenes. Furthermore, we compared our method with two existing advanced visual inertial navigation systems: VINS-Mono[11] and ORB-SLAM3[12]. Both VINS-Mono and ORB-SLAM3 are based on the nonlinear optimization framework and are currently recognized as excellent open-source visual inertial navigation systems. To verify the universality of the methods, we run the aforementioned three methods without considering loop closure. This is because, in many navigation applications, trajectories are unidirectional and do not involve loop closure. All the experiments were run on an Intel(R) Core (TM) i7 CPU with 3.60 GHz and 16 GB of RAM.

2.1 Experiments on public datasets

In this subsection, we performed the proposed VIO on the European Robotics Challenge (EuRoC) datasets[33]. The EuRoC datasets were captured by a binocular visual inertial sensor system mounted on an AscTec UAV (Unmanned aerial vehicle) platform, which provides synchronized binocular images and IMU data with precise timestamps. The binocular images

were captured at the frame rate of 20 Hz with the resolution of 752×480 pixels, and the acquisition frequency of IMU data was 200 Hz. We adopted left eye images and IMU data to construct a monocular visual inertial sensor system. The EuRoC datasets also provides ground truth captured by a Vicon motion capture system and a Leica MS50 laser for performance evaluation. Because of space limitations, we selected two representative scenes ("MH_05" and "V1_03") and presented the trajectory and absolute position error curves in Fig. 6 and 7, respectively. Moreover, the statistical results of position root mean square error (RMSE) on the whole EuRoC datasets are shown in Table 1.



(a) Trajectory curves (b) Absolute position error on axis X

(c) Absolute position error on axis Y (d) Absolute position error on axis Z

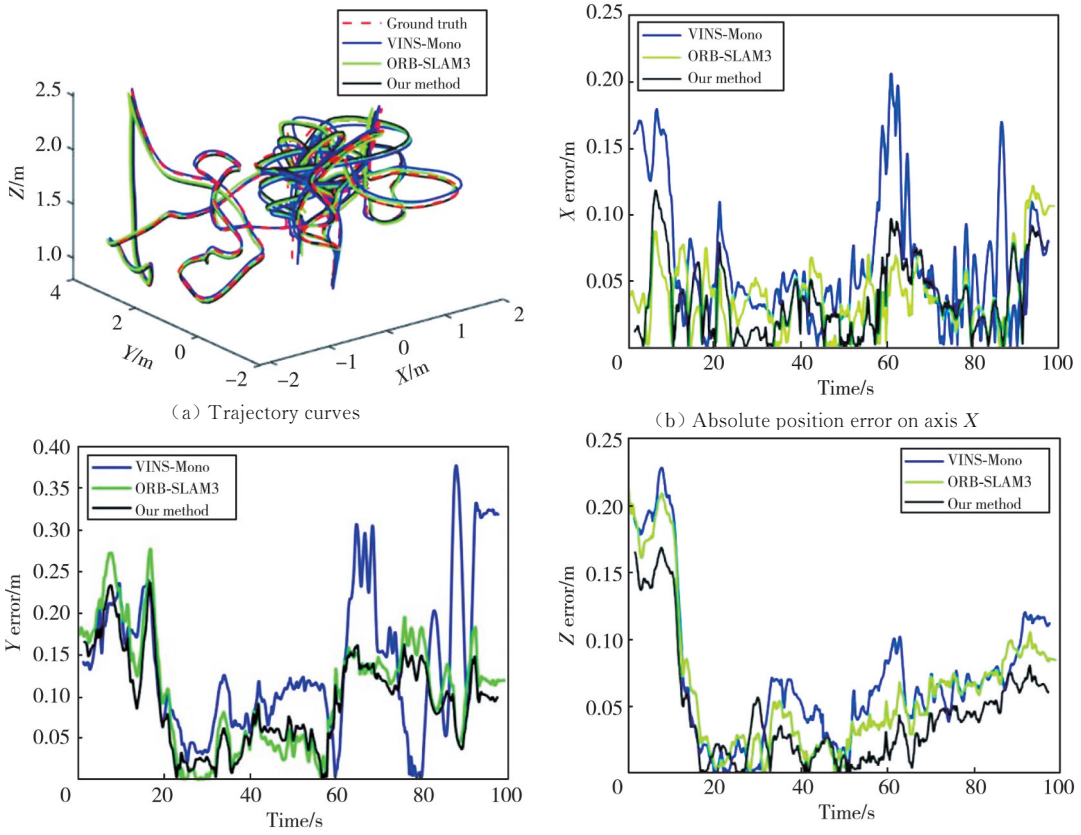
Fig. 6 Experimental results of three different methods on "MH_05" datasets

Table 1 Position RMSE of three different methods on EuRoC datasets (The best performance across each experiment and metric is highlighted in bold)

Datasets	Position RMSE/m		
	VINS-Mono [11]	ORB-SLAM3 [12]	Our method
MH_01	0.183	0.083	**0.064**
MH_02	0.241	**0.101**	0.112
MH_03	0.314	0.146	**0.087**
MH_04	0.272	0.119	**0.092**
MH_05	0.361	0.122	**0.090**
V1_01	0.167	**0.068**	0.071
V1_02	0.185	0.139	**0.115**
V1_03	0.202	0.158	**0.132**
V2_01	0.086	0.053	**0.036**
V2_02	0.199	0.110	**0.087**
V2_03	0.302	0.182	**0.137**
Average	0.228	0.116	**0.093**

It can be seen from Fig.6(a) and Fig.7(a) that VINS-Mono, ORB-SLAM3 and our method can all estimate the position effectively without significant drift. Furthermore, it can be seen from Table 1 that the average position RMSE of the above three algorithms on the EuRoC datasets is 0.228 m, 0.116 m and 0.093 m, respectively. Obviously, the positioning accuracy of our method is significantly better than that of the other two methods. This improvement is largely attributed to the utilization of planar environmental constraints and an adaptive robust optimization strategy. The planar environmental constraints can be transformed into observations, which increases the number of valid observations. Additionally, the adaptive robust optimization strategy assigns different weights to various feature points based on the layers in the image pyramid from which the feature points are extracted. This enhancement effectively improves the accuracy of the observations. In other words, compared with the other two algorithms, our method can provide more observations with higher accuracy. When in challenging environments, our method can provide more accurate

and robust state estimations. It simultaneously reduces the estimation error and eliminates short-term state drift caused by limited observations. It can also be seen that ORB-SLAM3 is more accurate than VINS-Mono on the EuRoC datasets. There are two main reasons for this. Firstly, ORB-SLAM3 uses ORB feature matching for tracking features, whereas VINS-Mono adopts the optical flow method. The accuracy and robustness of feature matching are better than those of optical flow. Secondly, VINS-Mono does not effectively process the point cloud data in the 3D map, whereas ORB-SLAM3 designs strategies for point cloud deletion, merging and optimization in the mapping thread, which enhances the accuracy to some extent.



(a) Trajectory curves

(b) Absolute position error on axis X

(c) Absolute position error on axis Y

(d) Absolute position error on axis Z

Fig. 7 Experimental results of three different methods on"V1_03" datasets

2.2 Experiments in outdoor scenes

In this subsection, vehicle-mounted navigation experiments were conducted in large-scale outdoor scenes. As shown in Fig. 8, an MYNTEYE-D1000-50 is fixed on a remote-controlled car to capture pictures with an oblique view. The MYNTEYE-D1000-50 integrates a compact binocular camera and an IMU. The binocular camera, with a baseline length of 120 mm, can provide synchronously triggered binocular images with a resolution of 640×360 (unit: pixel). It operated at a frame rate of 30 Hz, and the IMU data were acquired at a frequency of 200 Hz. We used left-eye images and IMU data to construct a monocular visual inertial sensor system and conducted outdoor experiments on the campus of Beihang University. The ground truth trajectories were provided by a PwrPak7-E1, which is a highly integrated navigation system using NovAtel SPAN technology.

Fig. 8 Data collection platform for our experiments in large-scale outdoor scenes: A MYNTEYE-D1000-50 (red part) is used to capture images and IMU data, and a Novatel PwrPak7D-E1 (yellow part) is used to provide ground truth trajectories

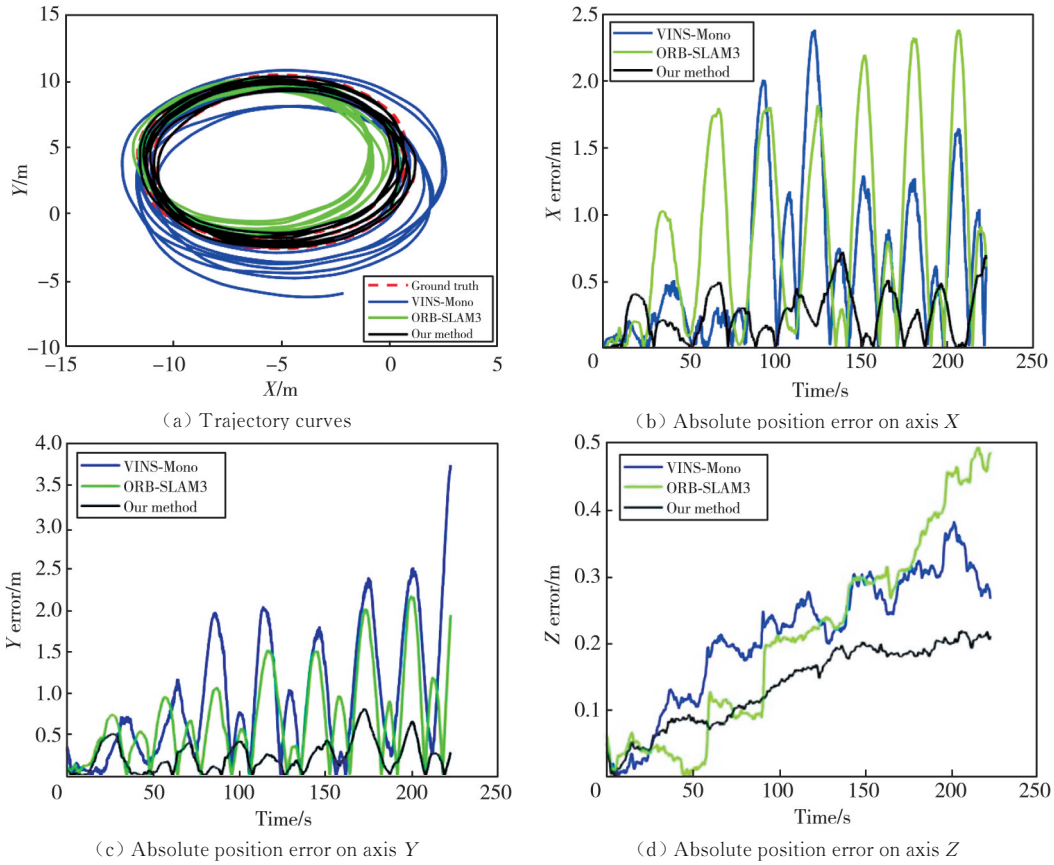
In order to evaluate the long-term applicability of the proposed algorithm in complex environments, three experiments with different trajectories were designed. In

experiment 1, we controlled the car to move in circles on the campus square. In experiment 2, we moved around the garden path in a cycle. In experiment 3, the car was controlled to move along the campus road in a rectangular trajectory. The lengths of the three trajectories were 309 m, 423 m and 1 787 m, respectively, and the durations of the three trajectories were 220 s, 517bs and 1 681 s, respectively. We present the trajectory and absolute position error curves in Figs.9—11. The statistical results of position RMSE are shown in Table 2.

It can be seen from Figs.9—11 the trajectory curves of VINS-Mono and ORB-SLAM3 have large error accumulations, resulting in significant drifts. However, our method can effectively estimate the carrier's position. In Table 2, the position RMSE of our method in the three experiments are 0.446 m, 1.187 m and 2.073 m, respectively, while the corresponding values for VINS-Mono are 1.517 m, 12.545 m and 14.596 m, and for ORB-SLAM3 are 1.410 m, 4.943 m and 9.423 m. Obviously, the positioning accuracy of the proposed method is significantly better than that of the other two methods in outdoor experiments. It can also be observed that, in our outdoor experiments, all three methods exhibit poorer positioning accuracy compared to that on the EuRoC datasets.

Table 2 Position RMSE of three different methods in outdoor experiments (The best performance across each experiment and metric is highlighted in bold)

Experiment	Position RMSE/m		
	VINS-Mono[11]	ORB-SLAM3[12]	Our method
Experiment 1	1.517	1.410	**0.446**
Experiment 2	12.545	4.943	**1.187**
Experiment 3	14.596	8.423	**2.073**

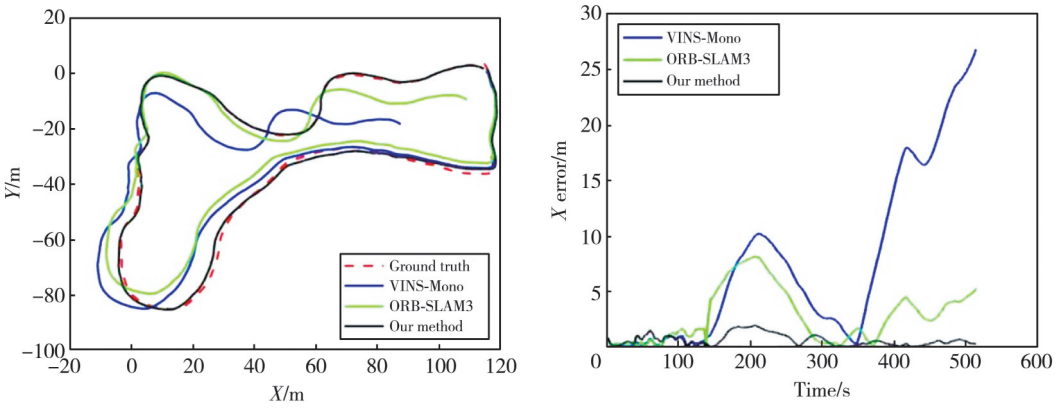


(a) Trajectory curves (b) Absolute position error on axis X

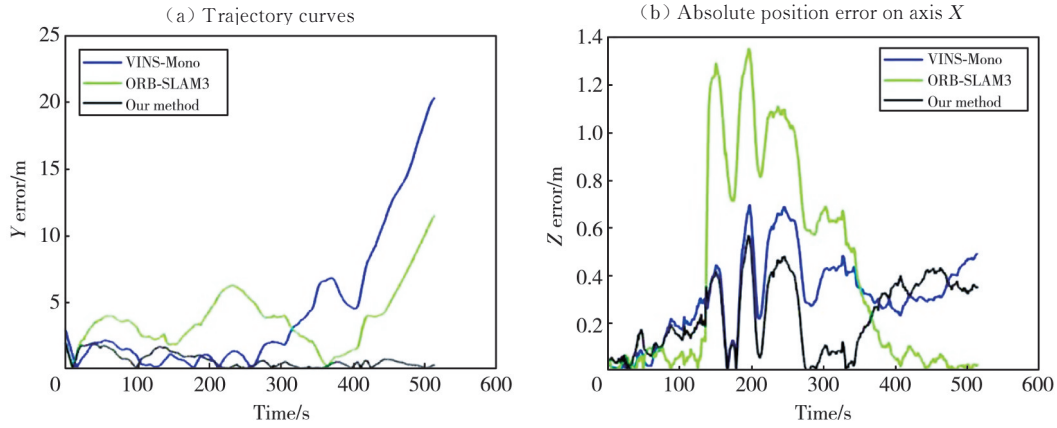
(c) Absolute position error on axis Y (d) Absolute position error on axis Z

Fig. 9 Experimental results of three different methods in experiment 1

This is mainly because our outdoor experiments have more complex environments and longer trajectories than the EuRoC datasets. The complex environments present significant challenges to the accuracy and stability of visual inertial navigation systems, and longer trajectories lead to more error accumulations in the experiments. Additionally, the cameras and IMU used in the EuRoC datasets are more accurate than the MYNTEYE-D1000-50. This is one of the reasons why the accuracy in our outdoor experiments is poorer than that in the EuRoC datasets. Even under challenging conditions, our algorithm is still capable of accurate positioning, which further indicates that our algorithm is highly adaptable to low-cost sensors and complex environments.

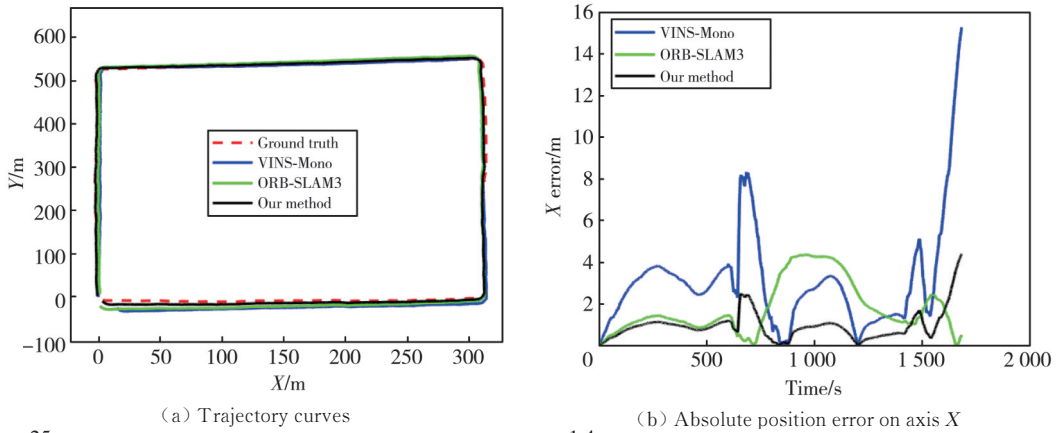


(a) Trajectory curves (b) Absolute position error on axis X

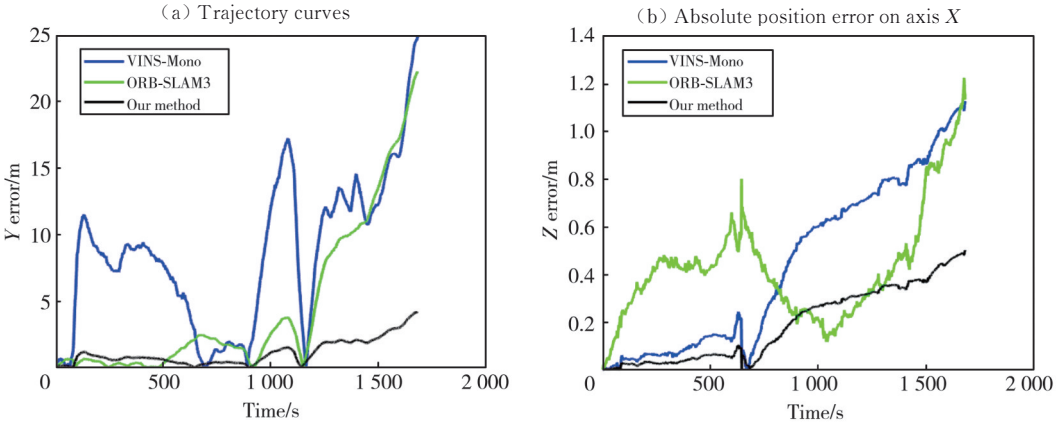


(c) Absolute position error on axis Y (d) Absolute position error on axis Z

Fig. 10 Experimental results of three different methods in experiment 2



(a) Trajectory curves (b) Absolute position error on axis X



(c) Absolute position error on axis Y (d) Absolute position error on axis Z

Fig. 11 Experimental results of three different methods in experiment 3

3 Conclusions

This paper introduces a novel monocular VIO that effectively utilizes planar environmental constraints, resulting in highly accurate and stable performance across wide dynamic, long-endurance and large-range navigation tasks. At the heart of our approach lies the integration of planar constraints into the visual inertial state estimation process through a nonlinear optimization framework. Furthermore, our method incorporates dynamic adjustments to error weights related to visual observations, further improving the system′s adaptability and precision during state estimation. Extensive experimental results consistently demonstrate the superiority of our system in achieving precise and stable navigation. As a result, our work holds significant potential for practical applications in achieving accurate and robust positioning across various domains, including robotics, autonomous vehicles and aerial systems.

Acknowledgement

This work was supported by Beijing Tongzhou District Science and Technology Innovation Talent Foundation (No. JCQN2023030); National Science Foundation of China (No.42274037); Aeronautical Science Foundation of China (No.2022Z022051001); Beijing Wuzi University Youth Research Foundation (No.2022XJQN22)

Declaration of conflicting interests

The authors have no conflict of interests related to this publication.

References

[1] DMMER G, BAUER H, Neumann R, et al. Design, additive manufacturing and component testing of pneumatic rotary vane actuators for lightweight robots. Rapid Prototyping Journal, 2022, 28(11): 20-32.

[2] CHEN M, WU Y, HE H. A novel navigation system for an autonomous mobile robot in an uncertain environment. Robotica, 2021, 40(3): 421-446.

[3] CUI G, LI B, TIAN W, et al. Dynamic modeling and vibration prediction of an industrial robot in manufacturing. Applied Mathematical Modelling, 2022, 105: 114-136.

[4] LI Z, ZHAO L, QIN C, et al. WiFi/PDR integrated navigation with robustly constrained Kalman filter. Measurement Science and Technology, 2020, 31(8): 84002.

[5] KOPPANYI Z, NAVRATIL V, XU H, et al. Using adaptive motion constraints to support UWB/IMU based navigation. Navigation, 2018, 65(2): 247-261.

[6] POULOSE A, HAN D. Hybrid indoor localization using IMU sensors and smartphone camera. Sensors, 2019, 19(23): 5084.

[7] GAO X, WANG R, DEMMEL N, et al. LDSO: Direct sparse odometry with loop closure//IEEE/RSJ International Conference on Intelligent Robots and Systems, October 1-5, 2018, Madrid, Spain. New York: IEEE, 2018: 2198-2204.

[8] LEE S, CIVERA J. Loosely-coupled semi-direct monocular SLAM. IEEE Robotics and Automation Letters, 2018, 4(2): 399-406.

[9] ZHANG M, CHEN Y, LI M. Vision-aided localization for ground robots//IEEE/RSJ International Conference on Intelligent Robots and Systems, November 4-8, 2019, Macau, China. New York: IEEE, 2019: 2455-2461.

[10] LEUTENEGGER S, LYNEN S, BOSSE M, et al. Keyframe-based visual-inertial odometry using nonlinear optimization. International Journal of Robotics Research, 2014, 34(3): 314-334.

[11] QIN T, LI P, SHEN S. VINS-mono: A robust and versatile monocular visual-inertial state estimator. IEEE Transactions on Robotics, 2018, 34(4): 1004-1020.

[12] CAMPOS C, ELVIRA R, JUAN J, et al. ORB-SLAM3: An accurate open-source library for visual, visual-inertial and multi-map SLAM. IEEE Transactions on Robotics, 2021, 37(6): 1874-1890.

[13] SUN K, MOHTA K, PFROMMER B, et al. Robust stereo visual inertial odometry for fast autonomous flight. IEEE Robotics and Automation Letters, 2018, 3(2): 965–972.

[14] KARAMAT T, LINS R, GIVIGI S, et al. Novel EKF-based vision/inertial system integration for improved navigation. IEEE Transactions on Instrumentation and Measurement, 2018, 67(1): 116-125.

[15] GENEVA P, ECKENHOFF K, LEE W, et al. OpenVINS: A research platform for visual-inertial estimation//IEEE International Conference on Robotics and Automation, May 31-June 15, 2020, Paris, France. New York: IEEE, 2020: 4666-4672.

[16] SHI X, LI D, ZHAO P, et al. Are we ready for service robots? The OpenLORIS-scene datasets for lifelong SLAM//IEEE International Conference on Robotics and Automation, May 31-June 15, 2020, Paris, France. New York: IEEE, 2020: 3139-3145.

[17] CHA J, JUNG J, CHUNG J, et al. Effect of wheel odometer on low-cost visual-inertial navigation system for ground vehicles//IEEE/ION Position, Location and Navigation Symposium, April 20-23, 2020, Portland, USA. New York: IEEE, 2020: 682-687.

[18] ZHAO H, JI X, WEI D, et al. Online IMU-odometer extrinsic calibration based on visual-inertial-odometer fusion for ground vehicles//IEEE International Conference on Indoor Positioning and Indoor Navigation, September 5-7, 2022, Beijing, China. New York: IEEE, 2022: 1-8.

[19] MAITY S, SAHA A, BHOWMICK B. Edge SLAM:

Edge points based monocular visual SLAM//IEEE International Conference on Computer Vision Workshops, October 27-29, 2017, Venice, Italy. New York: IEEE, 2017: 2408-2417.
[20] HE Y, ZHAO J, GUO Y, et al. PL-VIO: Tightly-coupled monocular visual-inertial odometry using point and line features. Sensors, 2018, 18(4):1159.
[21] HSIAO M, WESTMAN E, KAESS M. Dense planar-inertial SLAM with structural constraints//IEEE International Conference on Robotics and Automation, May 31-June 15, 2020, Paris, France. New York: IEEE, 2020: 6521-6528.
[22] GUO C, ROUMELIOTIS S. IMU-RGBD camera navigation using point and plane features//IEEE International Workshop on Intelligent Robots and Systems, November 3-7, 2013, Tokoyo, Japan. New York: IEEE, 2013: 3164-3171.
[23] ZUO X, YANG Y, GENEVA P, et al. LIC-Fusion 2.0: LiDAR-inertial-camera odometry with sliding-window plane-feature tracking//IEEE International Workshop on Intelligent Robots and Systems, May 31-June 15, 2020, Paris, France. New York: IEEE, 2020: 5112-5119.
[24] LEE W, YANG Y, HUANG G. Efficient multi-sensor aided inertial navigation with online calibration//IEEE International Conference on Robotics and Automation, May 30 -June 5, 2021, Xi′an, China. New York: IEEE, 2021: 5706-5712.
[25] YIN W, LIU Y, SHEN C. Virtual normal: Enforcing geometric constraints for accurate and robust depth prediction. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2022, 44(10): 7282-7295.
[26] LIU C, KIM K, GU J, et al. PlaneRCNN: 3D plane detection and reconstruction from a single image//IEEE/CVF Conference on Computer Vision and Pattern Recognition, June 15-29, 2019, Long Beach, CA, USA. New York: IEEE, 2019: 4450-4459.
[27] RAM K, KHARYAL C, HARITHAS S, et al. RP-VIO: Robust plane-based visual-inertial odometry for dynamic environments//IEEE International Workshop on Intelligent Robots and Systems, September 27-October 1, 2021, Prague, Czech Republic. New York: IEEE, 2021: 9198-9205.
[28] LI X, HE Y, LIN J, et al. Leveraging planar regularities for point line visual-inertial odometry//IEEE International Workshop on Intelligent Robots and Systems, May 31-June 15, 2020, Paris, France. New York: IEEE, 2020: 5120-5127.
[29] ROSTEN E, DRUMMOND T. Machine learning for high-speed corner detection//The 14th European Conference on Computer Vision, October 14-16, 2016, Amsterdam, The Netherlands. Berlin: Springer, 2016: 430-443.
[30] FORSTER C, PIZZOLI M, SCARAMUZZA D. SVO: Fast semi-direct monocular visual odometry//IEEE International Conference on Robotics and Automation, May 31- June 7, 2014, Hong Kong. China, New York: IEEE, 2014: 15-22.
[31] FORSTER C, CARLONE L, DELLAERT F, et al. On-manifold preintegration for real-time visual-inertial odometry. IEEE Transactions on Robotics, 2017, 33(1): 1-21.
[32] SIBLEY G, MATTHIES L, SUKHATME G. Sliding window filter with application to planetary landing. Journal of Field Robotics, 2010, 27(5): 587-608.
[33] BURRI M, NIKOLIC J, GOHL P, et al. The EuRoC micro aerial vehicle datasets. International Journal of Robotics Research, 2016, 35(10): 1157-1163.

平面环境约束辅助的单目视觉惯性里程计

多靖赟[1], 赵毅琳[2], 赵　龙[2], 李俊韬[1*]

1. 北京物资学院 智能物流系统北京市重点实验室, 北京 101149;
2. 北京航空航天大学 自动化科学与电气工程学院, 北京 100191

摘 要: 为提高视觉惯性导航系统在宽动态、长航时、大范围作业环境中的精确性与鲁棒性, 提出了一种平面环境约束辅助的单目视觉惯性里程计。通过在视频图像中提取并跟踪均匀分布的FAST特征点, 并采用对称光流剔除误跟踪点, 实现了视觉特征点高效检测与精确跟踪; 无需计算稠密深度地图, 仅从稀疏特征点集中检测共面特征点, 拟合空间平面, 构建了对视觉特征点三维坐标的空间几何约束; 融合视觉特征点重投影误差、共面特征点坐标约束以及惯性测量单元(Inertial measurement unit, IMU)预积分误差构造代价函数, 采用非线性优化方法估计了系统状态。最后, 分别在公开数据集和实际户外场景中评估了算法的准确性和有效性。实验结果表明, 相较于VINS-Mono和ORB-SLAM3算法, 本文方法的定位结果有显著提升, 实现了复杂应用环境中精确、稳定导航, 在机器人、无人驾驶等领域具有较大的实用价值。

关键词: 视觉惯性里程计; 平面环境约束; 状态估计; 非线性优化

引用格式: DUO Jingyun, ZHAO Yilin, ZHAO Long, et al. Planar environmental constraints aided monocular visual inertial odometry. Journal of Measurement Science and Instrumentation, 2024, 15(1): 83-94.