A Tightly-Coupled VIO Algorithm with GPS Assistance and Backend Pose Graph Optimization for Outdoor Applications

Wenlong Ji, Xu Guo, Anqi Xu, Siquan Li, Yanling Shang, and Fangzheng Gao

Abstract—**Traditional SLAM systems often suffer from drift accumulation and environmental ambiguity in outdoor settings, where varying lighting conditions and weather can severely degrade performance. To address these challenges, this paper introduces a novel GPS-enhanced visual-inertial SLAM framework that tightly integrates GPS with visual and inertial sensors at the front end while employing pose graph optimization at the back end for improved positioning accuracy and robustness. The proposed approach utilizes an Error-State Kalman Filter (ESKF) to fuse GPS, IMU, and visual data, significantly enhancing odometry precision. Furthermore, a sliding-window pose graph optimization technique is implemented in the back end to boost computational efficiency and maintain real-time performance. Experimental results on KITTI Sequences 07 and 06 demonstrate that our system achieves remarkable error reduction of 56.3% and 73.2%, respectively, compared to VIN-Fusion.**

Index Terms—**GPS-aided, outdoor localization, ESKF, Pose Graph Optimization.**

I. Introduction

SIMULTANEOUS Localization and Mapping (SLAM) is a fundamental technique in the field of robotics and autonomous systems, enabling a robot to simultaneously build a map of an unknown environment while estimating its own position within it, without relying on prior knowledge [1, 2]. Traditional SLAM methods, which predominantly rely on visual cues and inertial measurements, often encounter significant challenges in outdoor environments due to the dynamic and unpredictable nature of such settings. Variations in lighting, adverse weather conditions, and the absence of salient visual features can severely degrade the performance and robustness of visual SLAM systems [3–5]. In recent years, the integration of Global Positioning System (GPS) data into SLAM frameworks has emerged as a promising approach to enhance the robustness and accuracy of outdoor localization. GPS offers a global reference frame that helps mitigate drift and cumulative errors typically associated with traditional SLAM algorithms [6]. However, GPS signals are often unreliable or entirely unavailable in challenging environments such as urban canyons, tunnels, or areas with dense vegetation. To address these limitations, a hybrid localization strategy that fuses information from GPS, visual, and inertial sensors is essential for achieving reliable and precise positioning in complex outdoor scenarios.

In the domain of sensor integration, Xia et al. employed a visual–inertial sensor fusion framework augmented with global positioning observations from GNSS. These measurements were jointly optimized within a factor graph alongside local VIO poses. This multimodal and complementary approach demonstrated strong scalability and performed well in substation inspection tasks. However, the system was loosely coupled, leading to suboptimal utilization of sensor data. Similarly, R3live++ implemented separate LiDAR and VIO subsystems for geometric and photometric reconstruction, with integration achieved through back-end optimization. While effective, this architecture may result in information redundancy and feature-matching inconsistencies due to the absence of tightly-coupled, low-level joint optimization.

Factor graph optimization algorithms have been mainstream in the optimization process of the Visual-Inertial Odometry (VIO) back end in recent years. Factor graphs can flexibly express various constraints, such as sensor measurement constraints and motion model constraints [7]. However, factor graphs involve many factors. They need to handle multi-variable joint optimization and complex marginalization. This significantly increases computational complexity and poses challenges for real-time performance. Moreover, the approximation errors introduced during marginalization may degrade the overall robustness of the system. Therefore, it is essential to appropriately reduce the number of constraints in the factor graph to enhance computational efficiency and real-time capability.

In back-end factor graph optimization, Qin et al. proposed VIN-FUSION, which combines GPS measurements with VIO relative poses in a second optimization thread, integrating them into a general state estimation framework via graph optimization [8, 9]. Despite good practical results, it lacks online sensor calibration optimization. Similarly, Yao et al. introduced an adaptive fusion-based ground vehicle positioning method. It handles front-end data errors by merging residuals into the factor graph for nonlinear optimization. However, the complex factor graph-solving process consumes significant computational resources and

Manuscript received April 3, 2025; revised July 28, 2025. This work was partially supported by the National Natural Science Foundation of China under grant 61873120 and the Qing Lan project of Jiangsu Province.

Wenlong Ji is a postgraduate student in the School of Automation, Nanjing Institute of Technology, Nanjing 211167, China (e-mail:y00450240413@njit.edu.cn).

Xu Guo is a postgraduate student in the School of Automation, Nanjing Institute of Technology, Nanjing 211167, China (e-mail:y00450240314@njit.edu.cn).

Anqi Xu is a postgraduate student in the School of Automation, Nanjing Institute of Technology, Nanjing 211167, China (e-mail:y00450240304@njit.edu.cn).

Siquan Li is a postgraduate student in the School of Automation, Nanjing Institute of Technology, Nanjing 211167, China (email: y00450230122@njit.edu.cn).

Yanling Shang is an associate professor in the School of Automation, Nanjing Institute of Technology, Nanjing 211167, China (corresponding author, e-mail: hnnhsyl@126.com.com).

Fangzheng Gao is a professor in the School of Automation, Nanjing Institute of Technology, Nanjing 211167, China (e-mail:gaofz@126.com).

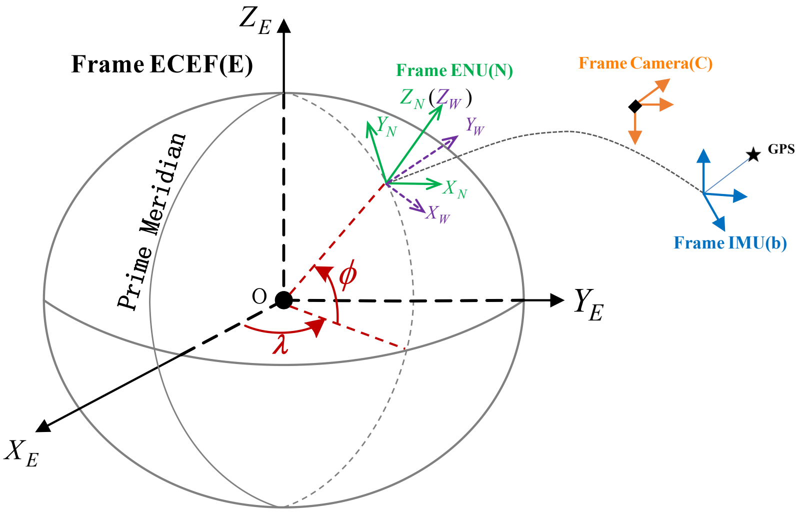


Fig. 1. Coordinate system diagram.

struggles to ensure real-time performance [10].

This paper proposes a novel GPS-aided visual-inertial SLAM system specifically designed to address the challenges of outdoor environments. By integrating GPS for global reference, visual sensors for local feature tracking, and IMU for robust motion estimation, our framework achieves superior performance through a synergistic fusion approach. The system employs a tightly-coupled front-end for sensor data fusion combined with an efficient back-end optimization strategy, offering significant improvements over conventional methods. The key contributions of this work include:

(1) A novel dual-optimization architecture that synergistically combines front-end ESKF with back-end pose graph optimization, overcoming the limitations of systems that rely solely on one optimization approach. This hybrid strategy provides more accurate and robust state estimation compared to the existing methods [11].

(2) An automatic calibration framework using ESKF to estimate and compensate for spatial-temporal offsets between GPS, IMU, and camera sensors. This innovation significantly enhances system robustness against sensor misalignment and synchronization errors.

(3) An efficient pose graph optimization method that simplifies traditional factor graph approaches by focusing on camera pose estimation while treating IMU biases as optional parameters. Combined with a sliding-window mechanism and incremental optimization, this approach maintains real-time performance without sacrificing accuracy.

The paper is organized as follows: Section II introduces the coordinate systems and mathematical notation. Section III details our algorithmic framework, including the ESKF-based front-end fusion and sliding-window pose graph optimization. Section IV presents experimental validation on real-world datasets, with comprehensive performance analysis. Finally, Section V concludes the paper and discusses potential future research directions.

II. Coordinate and Symbol Description

The coordinates involved in this paper are shown in Fig.1. The Earth-Centered, Earth-Fixed (ECEF) coordinate system is a global Cartesian coordinate system that is fixed relative to the Earth. The origin of the ECEF coordinate system is fixed at the Earth's center of mass. The X-Y plane coincides with the Earth's equatorial plane, with the X-axis pointing towards the Prime Meridian. The Z-axis is chosen to be perpendicular to the equatorial plane of the Earth, pointing towards the geographic North Pole.

This paper uses $\mathbf{R}_a^b$ and $\mathbf{p}_a^b$ to define the rotational and translational components of the transformation from coordinate system a to coordinate system b. The rotational component is represented by the corresponding Hamilton quaternion $\mathbf{q}_a^b$, where $\otimes$ denotes its multiplication operation. Subscripts are used to denote specific time points of the moving coordinate systems. The constant quantity g_W represents the gravity vector in the local world coordinate system, c is the speed of light in vacuum, ω_E represents the Earth's angular velocity, latitude λ, longitude ϕ, and geodetic height h. $\lfloor\cdot\rfloor\times$ represents the antisymmetric operator. ψ represents the yaw angle offset between the local world coordinate system(Local world) and the East-North-Up coordinate system (ENU), receiver clock offset δt, and receiver clock drift rate $\delta\dot{t}$. ρ represents the inverse depth of each feature.

III. Algorithm Framework

The system framework proposed in this paper is shown in Fig. 2, consisting of data preprocessing, GPS-aided tightly-coupled multi-sensor fusion measurement, and front-end filtering processing. The proposed ESKF-VIO algorithm employs an Error State Kalman Filter (ESKF) framework to achieve tight multi-sensor fusion through initial calibration. By integrating GPS measurements, IMU data, and visual observations within the ESKF, the system effectively combines onboard and external perception sources. This approach features continuous re-linearization of sensor constraints, significantly enhancing both odometric precision and attitude estimation accuracy in the front-end processing pipeline. Pose graph optimization in the back-end only processes pose nodes, with IMU bias as an optional element. A Sliding-Window mechanism is added, optimizing keyframe poses within the window while ignoring non-keyframe poses and other variables. This boosts computational efficiency and ensures real-time performance.

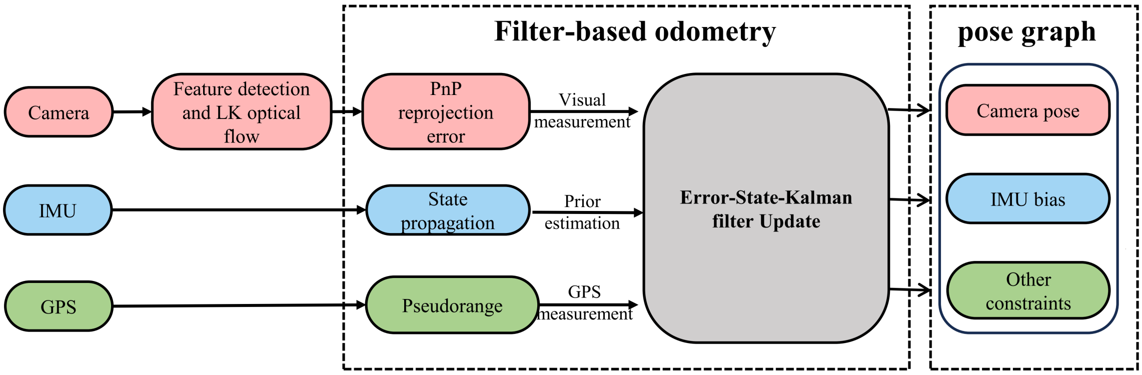


Fig. 2. System framework.

A. Feature tracking based on optical flow

At the core of our Visual-Inertial Odometry (VIO) front-end lies the Lucas-Kanade optical flow method, which provides efficient feature tracking across image sequences. Optical flow, representing the perceived motion of intensity patterns due to camera movement, captures both the direction and speed of pixel-level displacements [12]. This motion estimation forms the basis for our feature tracking mechanism, implemented through the following systematic approach:

Under the assumptions of brightness constancy, small motion, and spatial coherence, the intensity $I(x, y, t)$ of a pixel (x, y) in the image remains constant over time. Specifically, at time t, the pixel intensity is assumed to be a constant c, which leads to the derivation of the fundamental optical flow constraint equation.

$$I(x(t), y(t), t) = \mathrm{c} \tag{1}$$

Let the increments in spatial coordinates and time be denoted by d_x, d_y, and d_z, respectively. Then, we have:

$$I(x, y, t) = I\left(x + d_x, y + d_y, t + d_t\right) \tag{2}$$

Under the assumption of small motion, the position does not change significantly over time. Therefore, the image intensity function can be approximated by a first-order Taylor expansion at the location I(x, y, t) as follows:

$$\begin{aligned} &I(x, y, t) = I(x, y, t) + I_x d_x + I_y d_y + I_t d_t \\ &I_x u + I_y v + I_t = 0 \end{aligned} \tag{3}$$

where I_x and I_y denote the spatial image gradients in the horizontal and vertical directions, respectively, while I_z represents the temporal image gradient. $u = d_x/d_y$ and $v = d_y/d_x$ correspond to the horizontal and vertical components of the Optical Flow.

After extracting features using the Optical Flow method, feature matching and tracking are performed. The Optical Flow equations are solved using the least squares method to estimate the displacement of each pixel over a time interval t. The formulation is as follows:

$$\left[\begin{array}{c} u \\ v \end{array}\right] = \left[\begin{array}{cc} \sum\limits_{i=1}^{n} I_{ix}^{2} & \sum\limits_{i=1}^{n} I_{ix} I_{iy} \\ \sum\limits_{i=1}^{n} I_{ix} I_{iy} & \sum\limits_{i=1}^{n} I_{iy}^{2} \end{array}\right]^{-1} \left[\begin{array}{c} -\sum\limits_{i=1}^{n} I_{ix} I_t \\ -\sum\limits_{i=1}^{n} I_{iy} I_t \end{array}\right] \tag{4}$$

The results of the image tracking experiment are shown in Fig.3. Feature points with fewer successful tracking instances are marked with red solid dots, while those with more frequent successful tracking are marked with blue solid dots. Feature correspondences between the left and right images that are successfully tracked using the optical flow method are indicated by green solid dots. As illustrated in the figure, both edge features and texture-rich corner points are effectively tracked, with a low incidence of mismatches. By employing the Lucas-Kanade (LK) Optical Flow method, the stereo camera system successfully accomplishes robust feature tracking across image sequences.

B. ESKF-VIO design

An Error-State Kalman Filter (ESKF) for front-end odometry processing is employed to enable robust attitude estimation through tight multi-sensor fusion. The ESKF's fundamental innovation lies in its error-state formulation, which offers three key advantages: (1) the small magnitude of error states permits linear approximation while minimizing precision loss, (2) it simplifies Jacobian matrix computation by eliminating higher-order terms, and (3) it naturally supports rotation vector representation, effectively avoiding both parameter redundancy and singularity problems [13–15].

In this paper, let $\mathcal{M}$ be the manifold where the system state resides. The operations "$\boxplus$" and "$\boxminus$" on the manifold $\mathcal{M}$ are used to simplify representation and derivation. Since the manifold is locally homeomorphic to $\mathbb{R}^n$, where n is the dimension of $\mathcal{M}$, a bijective relationship can be established between the local neighborhood of $\mathcal{M}$ and the tangent space $\mathcal{M}$:

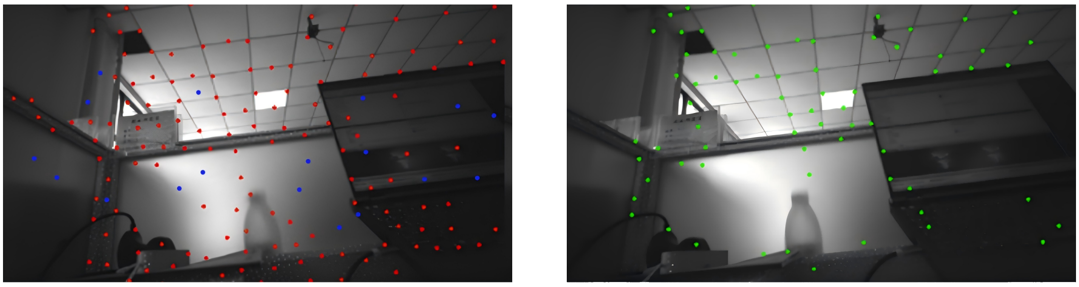
$$\boxplus : \mathcal{M} \times \mathbb{R}^n \rightarrow \mathcal{M}, \boxminus : \mathcal{M} \times \mathcal{M} \rightarrow \mathbb{R}^n \tag{5}$$

For the composite manifold $\mathcal{M} = SO(3) \times \mathbb{R}^n$, there is:

$$\begin{aligned} \begin{bmatrix} R \\ a_1 \end{bmatrix} \boxplus \begin{bmatrix} r \\ a_2 \end{bmatrix} &\triangleq \begin{bmatrix} R \cdot Exp(r) \\ a_1 + a_2 \end{bmatrix} \\ \begin{bmatrix} R_1 \\ a_1 \end{bmatrix} \boxminus \begin{bmatrix} R_2 \\ a_2 \end{bmatrix} &\triangleq \begin{bmatrix} Log(R_2^T R_1) \\ a_1 - a_2 \end{bmatrix} \end{aligned} \tag{6}$$

where $r \in \mathbb{R}^3$,$a_1, a_2 \in \mathbb{R}^3$,$Exp(\cdot)$ and $Log(\cdot)$denote the Rodriguez transformation between rotation matrices and rotation vectors. Besides, x_{k+1}, $\widetilde{x}_{k+1}$, δx_k respectively represents the true states, the nominal states, and the error states, and the Lie group is denoted by the uppercase letter 'G'.

a) Continuous time motion model: Assuming pre-calibration of the time drift between the camera, IMU, and GPS, factory-integrated calibration of the external parameters of GPS and IMU, and online estimation of the external



(a) left image

(b) right image

Fig. 3. Optical flow-based feature tracking.

parameters between the camera and IMU, the continuous motion model of Equation (7) is obtained using the IMU as the body frame:

$$\begin{aligned} \dot{p}_b^E &= v_b^E \\ \dot{v}_b^E &= R_b^E(a_m - b_a - n_a) + g^E \\ \dot{R}_b^E &= R_b^E[\omega_m - b_g - n_g] \\ \dot{b}_g &= n_{b_g} \\ \dot{b}_a &= n_{b_a} \end{aligned} \tag{7}$$

where R_b^E and p_b^E represent the orientation and position of the IMU relative to the global coordinate system, g^E represents the gravity acceleration, and ω_m and a_m represent the raw readings of the gyroscope and accelerometer. Assuming the three sensors are rigidly connected, with the extrinsic parameters between the camera and IMU denoted as $T_c^i = (R_c^i, p_c^i)$, representing the raw readings of the gyroscope and accelerometer, n_a and n_g represent the white noise of IMU measurements, b_a and b_g represent the biases of the accelerometer and gyroscope, modeled with Gaussian noise n_{b_g} and random walk n_{b_a}.

b) Discrete IMU model: Discretize the continuous model at the IMU frequency, and denote x_i as the state vector.

$$x_i = [{R_{b_i}^E}^T, {p_{b_i}^E}^T, {R_{c_i}^b}^T, {p_{c_i}^b}^T, {v_I^E}^T, {b_{g_i}}^T, {b_{a_i}}^T]^T \tag{8}$$

Using a zero-order hold to discretize, the discrete model is obtained:

$$x_{t+1} = x_i \boxplus (\Delta t f(x_i, u_i, w_i)) \tag{9}$$

Use the iterative ESKF to estimate the state vector, compute the state estimation error in the tangent space of the state manifold, and derive the propagation formula for the linearized error dynamics in the error-state space:

$$\begin{aligned} &\delta\hat{x}_{i+1} = x_{i+1} \boxminus \hat{x}_{i+1} \\ &= (x_i \boxplus (\Delta t \cdot f(x_i, u_i, w_i))) \\ &\boxminus (\hat{x}_i \boxplus (\Delta t \cdot f(\hat{x}_i, u_i, 0))) \sim N(0_{21\times 1}, \Sigma_{\delta\hat{x}_{i+1}}) \end{aligned} \tag{10}$$

where

$$\begin{aligned} \Sigma_{\delta\hat{x}_{i+1}} &= F_{\delta\hat{x}} \Sigma_{\delta\hat{x}_i} F_{\delta\hat{x}}^T + F_w Q F_w^T \\ F_{\delta\hat{x}} &= \frac{\partial(\delta\hat{x}_{i+1})}{\partial\delta\hat{x}_i}|_{\delta\hat{x}_i=0, w_i=0} \\ F_w &= \frac{\partial(\delta\hat{x}_{i+1})}{\partial w_i}|_{\delta\hat{x}_i=0, w_i=0} \end{aligned} \tag{11}$$

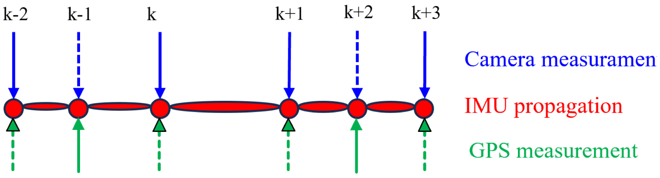


Fig. 4. Illustration of state iterated kalman filter update.

c) Prior Distribution.: Let the two propagation methods mentioned above stop at the $(k+1)_{th}$ GPS/Camera measurement as shown in Fig.4.

Then the propagated state estimate $\hat{x}_{k+1}$ and covariance $\Sigma_{\delta\hat{x}_{k+1}}$ essentially impose a prior distribution on the state before fusing the $(k+1)$-th measurement data, as follows:

$$x_{k+1} \boxminus \hat{x}_{k+1} \sim N(0, \Sigma_{\delta\hat{x}_{k+1}}) \tag{12}$$

d) Initialization of the iterative update.: The prior distribution is fused with GPS or camera measurement data to generate a maximum a posteriori (MAP) estimate x_{k+1} (denoted as $\widetilde{x}_{k+1}$). The MAP estimate $\widetilde{x}_{k+1}$) is initialized as the prior estimate $\hat{x}_{k+1}$, and due to the nonlinear nature of the problem, it is iteratively optimized. In each iteration, the error between the true state x_{k+1} and the current estimate $\widetilde{x}_{k+1}$ is defined as:

$$\delta\widetilde{x}_{k+1} \triangleq x_{k+1} \boxminus \widetilde{x}_{k+1} \tag{13}$$

The current state is obtained by minimizing the posterior distribution, which combines the prior in Equation (5) and the GPS/Camera measurement data. Therefore, the prior distribution represented by the term involving x_{k+1} should be transformed into an equivalent prior distribution involving $\delta\widetilde{x}_{k+1}$:

$$\begin{aligned} x_{k+1} \boxminus \hat{x}_{k+1} &= (\widetilde{x}_{k+1} \boxplus \delta\widetilde{x}_{k+1}) \boxminus \hat{x}_{k+1} \\ &\approx \widetilde{x}_{k+1} \boxminus \hat{x}_{k+1} + \mathcal{H}\delta\widetilde{x}_{k+1} \\ &\sim N(0, \Sigma_{\delta\hat{x}_{k+1}}) \end{aligned} \tag{14}$$

where, $\mathcal{H} = \frac{(\widetilde{x}_{k+1} \boxplus \delta\widetilde{x}_{k+1}) \boxplus \hat{x}_{k+1}}{\partial\delta\widetilde{x}_{k+1}}|_{\delta\widetilde{x}_{k+1}=0}$,the final expression for the prior distribution of $\delta\widetilde{x}_{k+1}$ is obtained. Equation (11) is derived as follows:

$$\delta\widetilde{x}_{k+1} \sim N(-\mathcal{H}^{-1}(\widetilde{x}_{k+1} \boxplus \hat{x}_{k+1}), \mathcal{H}^{-1}\Sigma_{\delta\hat{x}_{k+1}}\mathcal{H}^{-T}) \tag{15}$$

e) GPS measurement.: Utilize Broadcast Ephemeris Correction or compute Satellite Clock Bias and Orbit to correct Ionospheric and Tropospheric effects[16]. Employ PRN encoding to infer signal flight time offsets to obtain the expression for pseudo-range measurements:

$$\tilde{P}_r^S = \| \mathbf{p}_S^E - \mathbf{p}_r^E \| + c\left(\Delta t - \Delta t^s\right) + T_r^S + I_r^S + M_r^S + \epsilon_r^S \tag{16}$$

In the equation, $\mathbf{p}_S^E$ and $\mathbf{p}_r^E$ represent the Earth-Centered Inertial (ECI) coordinates of satellite S and receiver r, respectively. δt denotes the clock bias of GPS time, and Δt^s denotes the satellite clock offset, which can be obtained from ephemeris data. T_r^S and I_r^s represent the delay errors caused by the ionosphere and troposphere, respectively. M_r^S denotes the delay caused by multipath effects, and ϵ_r^S represents the measurement noise, which follows a Gaussian distribution with a mean of 0.

Through a series of coordinate transformations, model the pseudo-range measurement error factors for artificial satellites:

$$\begin{aligned} r_{\mathcal{P}}\left(\widetilde{\mathbf{z}}_{r_k}^{s_j}, \mathcal{X}\right) = & \left\| \mathbf{R}_z\left(-\omega_e t_f\right) \mathbf{p}_{s_j}^{e'} - \mathbf{R}_n^e \mathbf{R}_w^n \left(\mathbf{p}_b^w\right) - \mathbf{p}_{\text{anc}}^e \right\| \\ & + c\left(\zeta_{s_j}^T \delta \mathbf{t}_k - \Delta t^{s_j}\right) \\ & + T_{r_k}^{s_j} + I_{r_k}^{s_j} - \tilde{P}_{r_k}^{s_j} \end{aligned} \tag{17}$$

f) Visual measurement.: Extract FAST Corner Features from undistorted images and use the Lucas-Kanade (LK) Optical Flow to track feature points visible in the current sliding window by keyframes. If feature points are lost or not tracked, triangulate new feature points in 3D space as the best estimate of the camera pose. Update the current state estimation using the reprojection error between visual landmarks and tracked feature points, and derive the relevant formulas considering measurement noise:

$$P_s^E = P_s^{E^{gt}} + n_{P_s}, n_{P_s} \sim N(0, \Sigma_{n_{P_s}}) \tag{18}$$

$$p_s^c = p_s^{c^{gt}} + n_{p_s}, n_{p_s} \sim N(0, \Sigma_{n_{p_s}}) \tag{19}$$

where, $P_s^{E^{gt}}$ and $p_s^{c^{gt}}$ represent the true values of P_s^E and p_s^c, respectively. The first-order Taylor expansion of the true zero residual $r_c(x_{k+1}, p_s^{c^{gt}})$ is given by equation above, which forms another posterior distribution of $\delta\widetilde{x}_{k+1}$:

$$\begin{aligned} 0 &= r_c(x_{k+1}, p_s^{c^{gt}}, P_s^{E^{gt}}) \\ &\approx r_c(\widetilde{x}_{k+1}, p_s^c, P_s^E) + H_s^c \delta\widetilde{x}_{k+1} + \beta_s \end{aligned} \tag{20}$$

g) The update of the Error-State Kalman Filter.: Combining the prior distribution with the posterior distributions from the Camera/GPS measurements, we obtain the Maximum A Posteriori (MAP) estimate:

$$\begin{aligned} & min(\| \widetilde{x}_{k+1} \boxminus \hat{x}_{k+1} + \mathcal{H}\delta\widetilde{x}_{k+1} \|_{\Sigma^{-1}_{\delta\hat{x}_{k+1}}} \\ & + \sum_{j=1}^{m_l} \| r_E(\widetilde{x}_{k+1}, {}^L p_j) + H_j^E \delta\widetilde{x}_{k+1} \|^2_{\Sigma^{-1}_{\alpha_j}} \\ & + \sum_{s=1}^{m_c} \| r_c(\widetilde{x}_{k+1}, p_x^c, P_s^E) + H_s^c \delta\widetilde{x}_{k+1} \|^2_{\Sigma^{-1}_{\beta_s}}) \end{aligned} \tag{21}$$

where $\|x\|^2_\Sigma = x\Sigma x^T$.

Since GPS and Camera measurements may not occur at the same time, m_E (m_c) may be zero during the optimization process above, which indicates:

$$\begin{aligned} & H^T = [H_1^E, \ldots, H_{m_E}^E, H_1^c, \ldots, H_{m_c}^c]^T \\ & R = diag(\Sigma_{\alpha_1}, \ldots, \Sigma_{\alpha_{m_E}}, \Sigma_{\beta_1}, \ldots, \Sigma_{\beta_{m_c}}) \\ & P = (\mathcal{H})^{-1} \Sigma_{\delta\hat{x}_{k+1}} (\mathcal{H})^{-T} \\ & \widetilde{z}_{k+1}^T = [r_E(\widetilde{x}_{k+1}, p_1^E), \ldots, r_E(\widetilde{x}_{k+1}, p_{m_E}^E) \\ & r_c(\widetilde{x}_{k+1}, p_1^c, P_1^E), \ldots, r_c(\widetilde{x}_{k+1}, p_{m_c}^c, P_{m_c}^E)] \end{aligned} \tag{22}$$

The Kalman gain is obtained as follows:

$$K = (H^T R^{-1} H + P^{-1})^{-1} H^T R^{-1} \tag{23}$$

To update the state estimate, Equation (24) is given as follows:

$$\widetilde{x}_{k+1} = \widetilde{x}_{k+1} \boxplus (-K\widetilde{z}_{k+1} - (I - KH)(\mathcal{H})^{-1}(\widetilde{x}_{k+1} \boxminus \hat{x}_{k+1})) \tag{24}$$

The above process is iterated until convergence.

$$\begin{aligned} \hat{x}_{k+1} &= \widetilde{x}_{k+1} \\ \hat{\Sigma}_{\delta\bar{x}_{k+1}} &= (I - KH)\widetilde{\Sigma}_{\delta x_{k+1}} \end{aligned} \tag{25}$$

C. Pose graph optimization

The main computational burden of BA (Bundle Adjustment) optimization comes from feature point optimization. As feature points increase, computational efficiency decreases. In reality, after several observations, converged feature points' spatial position estimates remain near a certain value, while divergent outliers usually disappear.

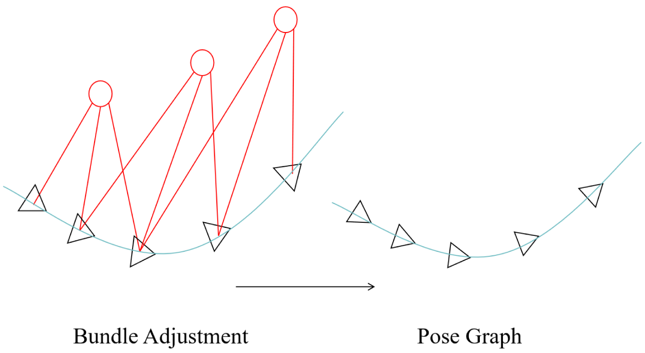


Fig. 5. Principle of pose graph optimization.

Based on this, we assume that a feature point can be ignored after a few optimizations, treating it only as a constraint for pose estimation, without further optimizing its position. The principle is shown in Fig.5. This creates a graph optimization with only trajectories and no feature points, where nodes are posed at different times, and edges are initial values from feature matching between adjacent frames [17, 18].

Unlike BA optimization, once initial estimates are done, we no longer optimize feature point positions, focusing only on connections between camera poses. This significantly reduces the computation.

In pose graph problems, each node represents the camera's current pose using its Lie algebra. Edges represent the estimated relative motion between adjacent frames' pose nodes, from either direct methods or feature point methods. In SLAM frameworks, other constraints, such as loop closure

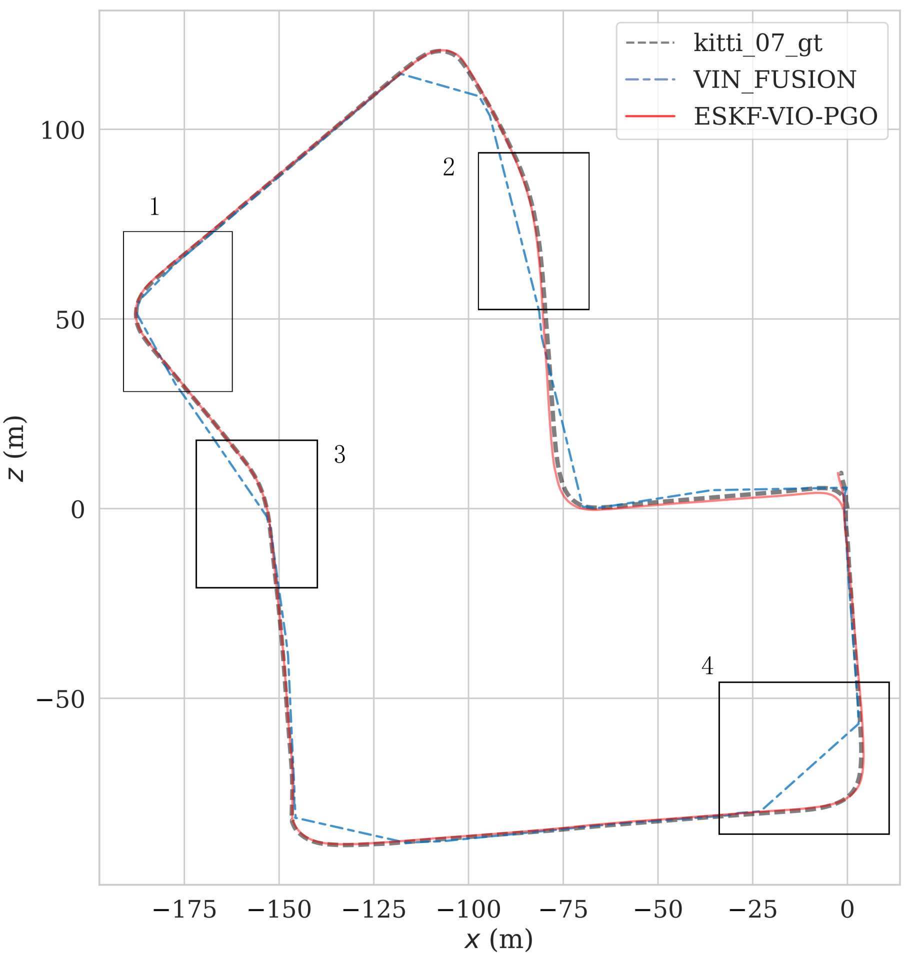


Fig. 6. Simulation trajectory comparison.

detection, may also exist. This motion can be expressed in terms of Lie groups:

$$\boldsymbol{T}_{ij} = \boldsymbol{T}_i^{-1}\boldsymbol{T}_j \tag{26}$$

From a graph optimization perspective, this equation has errors. So we can define a least squares error and use optimization methods to find the optimal solution. The error can be defined as follows:

$$\begin{aligned} \boldsymbol{e}_{ij} &= \ln\left(\Delta\boldsymbol{T}_{ij}^{-1}\boldsymbol{T}_i^{-1}\boldsymbol{T}_j\right)^{\vee} \\ &= \ln\left(\exp\left((-\boldsymbol{\xi}_{ij})^{\wedge}\right)\exp\left((-\boldsymbol{\xi}_i)^{\wedge}\right)\exp\left(\boldsymbol{\xi}_j^{\wedge}\right)\right)^{\vee} \end{aligned} \tag{27}$$

In the error definition equation, the relative pose can be obtained using algorithms with IMU and camera sensors, so they are known. Following the Lie algebra derivative method, we apply a left perturbation, and the error becomes:

$$\hat{\boldsymbol{e}}_{ij} = \ln\left(\boldsymbol{T}_{ij}^{-1}\boldsymbol{T}_i^{-1}\exp\left((-\delta\boldsymbol{\xi}_i)^{\wedge}\right)\exp\left(\delta\boldsymbol{\xi}_j^{\wedge}\right)\boldsymbol{T}_j\right)^{\vee}. \tag{28}$$

In this formula, the two perturbation items are in the middle. To use the BCH approximation, move and slightly change the perturbation items to get the following equation:

$$\exp\left(\boldsymbol{\xi}^{\wedge}\right)\boldsymbol{T} = \boldsymbol{T}\exp\left(\left(\mathrm{Ad}\left(\boldsymbol{T}^{-1}\right)\boldsymbol{\xi}\right)^{\wedge}\right) \tag{29}$$

Derive the Jacobian matrix in the right-multiplication form.

$$\begin{aligned} \hat{\boldsymbol{e}}_{ij} &= \ln\left(\boldsymbol{T}_{ij}^{-1}\boldsymbol{T}_i^{-1}\exp\left((-\delta\boldsymbol{\xi}_i)^{\wedge}\right)\exp\left(\delta\boldsymbol{\xi}_j^{\wedge}\right)\boldsymbol{T}_j\right)^{\vee} \\ &\approx \boldsymbol{e}_{ij} + \frac{\partial \boldsymbol{e}_{ij}}{\partial \delta\boldsymbol{\xi}_i}\delta\boldsymbol{\xi}_i + \frac{\partial \boldsymbol{e}_{ij}}{\partial \delta\boldsymbol{\xi}_j}\delta\boldsymbol{\xi}_j \end{aligned} \tag{30}$$

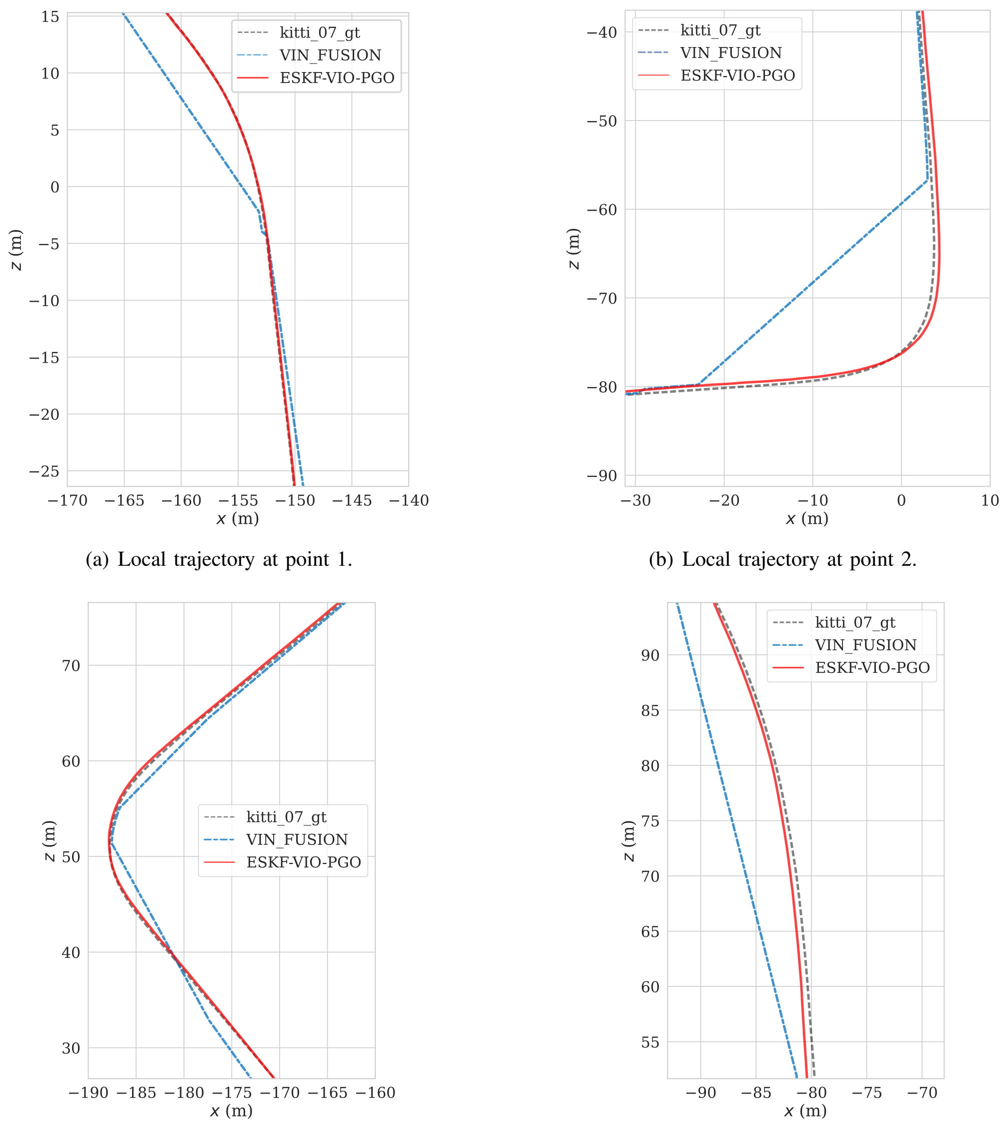
Then derive the Jacobian matrix and define the objective function for solution with graph optimization. The overall objective function is:

$$\min_{\boldsymbol{\xi}} \frac{1}{2}\sum_{i,j\in\mathcal{E}} \boldsymbol{e}_{ij}^T\boldsymbol{\Sigma}_{ij}^{-1}\boldsymbol{e}_{ij} \tag{31}$$

In short, all pose vertices and pose edges form a graph optimization, with optimization variables as poses of respective vertices and edges from pose observation constraints. It's transformed into a least squares problem for a solution. A Sliding-Window mechanism is introduced to reduce computation. The sliding window limits keyframes in the optimization window, lowering computational complexity for local real-time optimization. Pose graphs trigger global optimization periodically in the background to correct accumulated errors within the window. Together, they enhance computational efficiency and algorithm accuracy through marginalization and relocalization [16].

IV. Simulation verification

Simulation experiments were conducted using the KITTI dataset to verify the performance of the algorithm presented in this paper. The KITTI dataset, sponsored jointly by the Karlsruhe Institute of Technology and the Toyota Technological Institute in Chicago, is a dataset used for autonomous driving research. It contains various modalities of information, such as calibrated and synchronized images, lidar scans,



(a) Local trajectory at point 1.

(b) Local trajectory at point 2.

(c) Local trajectory at point 3.

(d) Local trajectory at point 4.

Fig. 7. Local trajectory at different points in KITTI_07.

high-precision GPS data, and IMU acceleration data, making it suitable for the testing and simulation environment of this paper. To validate the effectiveness of the method proposed in this paper, algorithm evaluation tests were performed using the KITTI 07 and 06 datasets as representative examples. Each KITTI sequence is recorded at 10 Hz and includes synchronized grayscale stereo and color images. All images have a resolution of 1392×512 pixels, and rectified stereo pairs are available.

Sequence 06 contains approximately 4541 frames, covering around 7.5 minutes of driving. The environment is mainly rural, including winding mountain roads, open fields, and some residential areas. Roads are narrow with dense vegetation on both sides. Tree occlusions and varying lighting conditions are common.

Sequence 07 includes about 4661 frames, covering around 7.8 minutes. It mixes urban and suburban roads, including intersections, roundabouts, pedestrian zones, and sections of highway. The environment includes traffic signs, signal lights, and moving vehicles. These scenes are suitable for testing algorithm robustness in complex interactions. The algorithm experimental environment is ROS Melodic; the operating system is Ubuntu 18.04, with 16GB of memory.

In Fig.6 and Fig.7, the black dashed line represents the actual ground trajectory generated by the KITTI_07 dataset. The blue dotted line indicates the trajectory graph generated by the VIN-Fusion algorithm, and the red solid line indicates the global trajectory graph generated by the ESKF-VIO-PGO algorithm proposed in this paper. The algorithm presented in this paper provides more accurate localization with trajectories that better fit the true values.

To evaluate trajectory accuracy errors accurately, we use the EVO tool to measure precisely and generate heat maps of trajectory errors for different algorithms. The solid line transitions from cold to warm colors, indicating increasing errors between the estimated and true trajectories. The right-side gradient bar shows the error sizes corresponding to the line colors, with maximum, average, and minimum errors provided on the right. The global trajectory error values for each algorithm extracted by EVO are in Table 1.

According to the data in Table 1, the final proposed GPS-aided visual-inertial odometry method improves posi-

TABLE I
TRAJECTORY ERROR VALUES

	Algorithm	Max/m	Min/m	Mean/m	RMSE/m	STD/m
KITTI-07	VIN-FUSION	8.914	0.022	4.118	5.155	3.101
	ESKF-VIO-PGO	4.821	0.135	1.801	2.231	1.316
KITTI-06	VIN-FUSION	4.576	0.194	2.652	3.009	1.421
	ESKF-VIO-PGO	3.444	0.131	0.448	0.645	0.464

tioning accuracy by 56.3% on the KITTI_07 dataset and 73.2% on the KITTI_06 dataset compared to traditional algorithms. The front-end ESKF-based tight-coupling algorithm effectively improves the utilization of multi-sensor data and enhances outdoor positioning accuracy. The back-end pose graph optimization reduces accumulated errors, further enhancing positioning accuracy and system robustness, and also boosts system efficiency.

V. Conclusion

To address the challenges of outdoor localization, this paper presents a tightly coupled GPS-aided visual-inertial odometry (VIO) algorithm. The front-end processing employs an ESKF to tightly fuse multi-sensor data, effectively mitigating odometric drift. Meanwhile, the back-end incorporates sliding-window pose graph optimization, ensuring both computational efficiency and enhanced system accuracy and robustness. Experimental results demonstrate that our method significantly outperforms the conventional VIN-Fusion algorithm in localization precision. By synergistically leveraging GPS for global reference, IMU for motion estimation, and visual data for feature tracking, our system delivers a highly reliable solution for outdoor robotic navigation. Future research will focus on algorithmic refinement and extending the framework to diverse real-world outdoor applications.

References

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

[2] Z. Hu, F. Li, and J. Shen, “A semantic slam integrated with enhanced yolov7 target detection algorithm,” *Engineering Letters*, vol. 32, no. 10, pp. 1909–1920, 2024.

[3] X. Wang, Y. Yao, J. Wen, J. Lin, D. Chen, Y. Cai, S. Li, Y. Shang, and F. Gao, “Indoor localization method enhanced by an improved harris’s hawk optimization based particle filter,” *IAENG International Journal of Computer Science*, vol. 53, no. 3, pp. 267–275, 2024.

[4] Y. Luo, J. Shen, and F. Li, “Real-time visual slam based on lightweight pspnet network,” *Engineering Letters*, vol. 32, no. 10, pp. 1981–1992, 2024.

[5] H. Pu, J. Luo, G. Wang, T. Huang, H. Liu, and J. Luo, “Visual slam integration with semantic segmentation and deep learning: a review,” *IEEE Sensors Journal*, vol. 23, no. 19, pp. 22 119–22 138, 2023.

[6] A. El-Rabbany and A. Kleusberg, “Effect of temporal physical correlation on accuracy estimation in gps relative positioning,” *Journal of Surveying Engineering*, vol. 129, no. 1, pp. 28–32, 2003.

[7] X. Wang, F. Gao, J. Huang, and Y. Xue, “Uwb/lidar tightly coupled positioning algorithm based on issa optimized particle filter,” *IEEE Sensors Journal*, vol. 24, no. 7, pp. 11 217–11 228, 2024.

[8] T. Qin, S. Cao, J. Pan, and S. Shen, “A general optimization-based framework for global pose estimation with multiple sensors,” *arXiv preprint arXiv:1901.03642*, 2019.

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

[10] Y. Zhang, L. Chu, Y. Mao, X. Yu, J. Wang, and C. Guo, “A vision/inertial navigation/global navigation satellite integrated system for relative and absolute localization in land vehicles,” *Sensors*, vol. 24, no. 10, p. 3079, 2024.

[11] Z. Lan, J. Wang, Z. Shen, and Z. Fang, “Highly robust and accurate multi-sensor fusion localization system for complex and challenging scenarios,” *Measurement*, vol. 235, p. 114851, 2024.

[12] A. Plyer, G. Le Besnerais, and F. Champagnat, “Massively parallel lucas-lanade optical flow for real-time video processing applications,” *Journal of Real-Time Image Processing*, vol. 11, pp. 713–730, 2016.

[13] M. Li, H. Zhu, S. You, and C. Tang, “Uwb-based localization system aided with inertial sensor for underground coal mine applications,” *IEEE Sensors Journal*, vol. 20, no. 12, pp. 6652–6669, 2020.

[14] J. Sola, “Quaternion kinematics for the error-state kalman filter,” *arXiv preprint arXiv:1711.02508*, 2017.

[15] N. Shaukat, A. Ali, M. Javed Iqbal, M. Moinuddin, and P. Otero, “Multi-sensor fusion for underwater vehicle localization by augmentation of rbf neural network and error-state kalman filter,” *Sensors*, vol. 21, no. 4, p. 1149, 2021.

[16] L. Zhou, J. Qiu, P. Han, D. Liu, and K. Luo, “Lidar pose estimation method based on factor graph optimization,” *Journal of Physics: Conference Series*, vol. 2816, no. 1, p. 012011, 2024.

[17] S. Teresa, L. María, E. Jesús, M. Luis, and V. José, “Robot localization in tunnels: combining discrete features in a pose graph framework,” *Sensors*, vol. 22, no. 4, p. 1390, 2022.

[18] Z. Zeng, L. Cui, M. Qian, Z. Zhang, and K. Wei, “A survey on sliding window sketch for network measurement,” *Computer Networks*, vol. 226, p. 109696, 2023.