miniVIO: A Minimalist Visual-Inertial Odometry Algorithm with Minimally Inferred Motion Constraints

Yuxiang Peng[1], Chuchu Chen[2], Tong Ke[3], Ryan C. DuToit[3], Shuntaro Yamazaki[3], Huiwen Guo[3], and Guoquan Huang[1]

***Abstract*— Visual-inertial odometry (VIO) algorithms prevail but are still computationally demanding for resource-constrained platforms due to their reliance on sliding windows of historical poses or maintaining many landmarks. To overcome these bottlenecks, in this work, we design a super-fast minimalist VIO framework (i.e., miniVIO) that operates only on a minimal set of navigation states. In particular, we derive a novel inferred motion constraint about local velocity and gravity from multi-frame (as few as two) visual feature tracks. As this inferred constraint depends only on the current navigation state, it can be immediately used to perform EKF update, without delay. Extensive real-world experiments on the public datasets demonstrate that our miniVIO preserves competitive estimation accuracy while achieving over an order-of-magnitude speedup compared to SOTA baselines. It executes at just 1.27 ms per frame even on a legacy Jetson Nano and maintains robust tracking even with camera reduced to 1 fps.**

I. Introduction and Related Work

Computationally efficient 3D motion tracking is essential for spatial awareness in autonomous robotics and extended reality (XR) applications. While state-of-the-art (SOTA) visual–inertial odometry (VIO) algorithms are able to provide robust and accurate motion estimates by fusing camera and IMU measurements [1], they remain computationally expensive for resource-constrained platforms such as micro aerial vehicles (MAVs) and AR glasses, particularly during continuous, always-on operations.

From the estimation perspective, existing VIO systems typically control computational complexity through sliding-window optimization or recursive filtering. Sliding-window optimization restricts the estimation problem to a bounded set of recent states and measurements at the cost of discarding or approximating past information through marginalization thus has a sub-optimal solution. Extensive research has therefore focused on state marginalization [2]–[5], inertial preintegration [6]–[8], incremental smoothing [9]–[11], measurement management and keyframe selection [12], as well as information sparsification [13] to further accelerate windowed optimization while preserving accuracy.

Alternatively, recursive filtering approaches, such as the Multi-State Constraint Kalman Filter (MSCKF) [14], avoid the immense computational burden of maintaining explicit 3D landmark states. Instead, MSCKF converts visual feature observations across a short history into motion constraints on cloned camera/IMU poses via nullspace projection or linear marginalization [5], [15]. Despite this innovation, MSCKF-like methods still require the state to be augmented with a sliding window of past poses to express multi-frame geometry. Consequently, the state dimension and update costs scale significantly with the number of maintained clones and feature tracks. Other approaches have attempted to compress visual information into low-dimensional motion cues, such as estimating local velocity from inter-frame feature motion [16], [17] or utilizing event-camera "velometers" [18]. While these highlight the value of compact motion constraints, they generally rely on external scale sources (like depth sensors) or serve as supplementary velocity modules rather than replacing multi-frame geometric constraints within the core estimator.

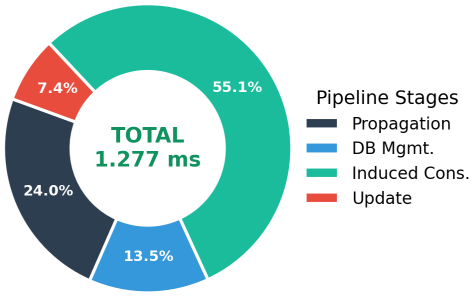

Fig. 1: The runtime composition of the proposed miniVIO running on a legacy Jetson Nano with ARM Cortex-A57 @1.5GHz. It is composed by four major blocks: propagation, update, database management and inferred motion constraint generation. To the best of our knowledge, this is the fastest VIO performance ever reported.

To further speed up the MSCKF-like VIO computation, we seek to infer motion constraints from visual measurements that directly update the current navigation state *without* augmenting the estimator with additional poses or landmark variables. Recognizing that even a short visual feature track (not necessarily consecutive) is able to provide robust and accurate translational direction information, leveraging IMU kinematics available in the VIO system, we infer a novel *inferred motion constraint* that relates the local velocity and gravity and can be used to, *immediately, without any delay*, update the current state. Importantly, because this constraint is expressed directly in terms of the current navigation states, our formulation effectively eliminates the need to maintain historic poses (stochastic clones) within the estimator state. Consequently, constraints can be constructed using arbitrary historical frames, provided that sufficient visual overlap exists with the current frame. Our main contributions are:

- Aided by inertial preintegration, we derive a novel in-

[1]The authors are with the Robot Perception and Navigation Group (RPNG), University of Delaware, Newark, DE 19716, USA. Email: {yxpeng, ghuang}@udel.edu.

[2]The author is with George Washington University, Washington, DC 20052, USA. Email: chuchu.chen@gwu.edu.

[2]The authors are with Google, Mountain View, CA 94043. Email: {tongk, rdutoit, shuntaro, calvinguo}@google.com.

ferred motion constraint about local velocity and gravity from multi-frame (as few as two) visual feature tracks. As this inferred constraint depends only on the current navigation state, it can be immediately used to perform EKF update, without delay.
- Based on our inferred inferred motion measurements, we develop a minimalist VIO algorithm that maintains only a minimum size of current navigation states, as few as 9DoF (position, rotation and velocity), and performs extremely lightweight update, achieving super fast performance (e.g., see Fig. 1).
- We demonstrate through extensive experiments that the proposed miniVIO preserves competitive estimation accuracy while significantly reducing computational overhead, achieving over an order-of-magnitude speedup over the fastest SOTA VIO baseline and maintaining robust performance under reduced update rates.

II. Problem Statement

In this section, we briefly describe arguably one of the most efficient VIO frameworks—MSCKF [14], [19]—which serves as the base framework for the proposed miniVIO; and then explain its computational challenges still facing to VIO in practice, which motivates the design of our miniVIO.

A. MSCKF-based VIO

A VIO algorithm aims to estimate the motion of the sensor platform by fusing inertial measurements with visual observations. Regardless computational cost, an optimal approach would be to jointly estimate the entire trajectory of states together with the observed scene structure given all available measurements. However, such a full-batch formulation can quickly become computationally infeasible in practice, and thus a filtering-based formulation such as the well-known MSCKF [14] is often used, by maintaining only the current window of states while marginalizing the past.

Specifically, at time t_k, the system state $\mathbf{x}_k$ consists of the current navigation state $\mathbf{x}_{I_k}$, historical IMU pose clones $\mathbf{x}_C$, and SLAM features $\mathbf{x}_f$:

$$\mathbf{x}_k = \begin{bmatrix}\mathbf{x}_{I_k}^\top & \mathbf{x}_C^\top & \mathbf{x}_f^\top\end{bmatrix}^\top \tag{1}$$

$$\mathbf{x}_{I_k} = \begin{bmatrix}{}^{I_k}_{G}\bar{q}^\top & {}^G\mathbf{p}_{I_k}^\top & {}^G\mathbf{v}_{I_k}^\top & \mathbf{b}_g^\top & \mathbf{b}_a^\top\end{bmatrix}^\top \tag{2}$$

$$\mathbf{x}_C = \begin{bmatrix}\mathbf{x}_{T_k}^\top \dots \mathbf{x}_{T_{k-c}}^\top\end{bmatrix}^\top, \quad \mathbf{x}_f = \begin{bmatrix}\mathbf{f}_1^\top \dots \mathbf{f}_g^\top\end{bmatrix}^\top \tag{3}$$

where ${}^{I_k}_{G}\bar{q}$ is the unit quaternion (${}^{I_k}_{G}\mathbf{R}$ in rotation matrix form) that represents the rotation from the global frame $\{G\}$ to the IMU frame $\{I_k\}$. ${}^G\mathbf{p}_{I_k}$ and ${}^G\mathbf{v}_{I_k}$ denote the IMU position and velocity expressed in $\{G\}$. $\mathbf{b}_g$ and $\mathbf{b}_a$ are the gyroscope and accelerometer biases. $\mathbf{f}_i$ denotes the i-th feature position, and $\mathbf{x}_{T_i} = [{}^{I_i}_{G}\bar{q}^\top \ {}^G\mathbf{p}_{I_i}^\top]^\top$ represents the i-th cloned IMU pose. The cloned poses $\mathbf{x}_C$ store a sliding window of historical camera poses, which enables the formation of multi-view geometric constraints while avoiding explicit estimation of feature states.

The inertial kinematics are used to evolve the state from time t_k to t_{k+1}:

$$\mathbf{x}_{I_{k+1}} = f(\mathbf{x}_{I_k}, \mathbf{a}_{m_k}, \boldsymbol{\omega}_{m_k}) \tag{4}$$

where the linear acceleration $\mathbf{a}_{m_k}$ and the angular velocity $\boldsymbol{\omega}_{m_k}$ measurements are contaminated by zero-mean white Gaussian noises. The nonlinear IMU kinematic model is linearized to propagate the state estimate and covariance forward [14].

The camera provides bearing observations of environmental 3D points. These observations can be used to update our state using the following measurement function (note that we here assume the global 3D feature model [19]):

$$\mathbf{z}_k = \mathbf{h}(\mathbf{x}_k) + \mathbf{n}_k =: \boldsymbol{\Lambda}({}^{C_k}\mathbf{f}) + \mathbf{n}_k \tag{5}$$

$${}^{C_k}\mathbf{f} = [x \ y \ z]^\top = {}^C_I\mathbf{R}\,{}^{I_k}_{G}\mathbf{R}\left({}^G\mathbf{f} - {}^G\mathbf{p}_{I_k}\right) + {}^C\mathbf{p}_I \tag{6}$$

$$\boldsymbol{\Lambda}([x \ y \ z]^\top) =: [x/z \ y/z]^\top \tag{7}$$

where $\mathbf{n}_k$ is the white Gaussian bearing measurement noise and $\{{}^C_I\mathbf{R}, {}^C\mathbf{p}_I\}$ is the camera-IMU rigid transformation. The standard MSCKF linearizes the measurement model and eliminates the feature states by projecting the stacked residuals onto the left nullspace of the feature Jacobian. However, the resulting constraint still depends on the cloned poses that observe the feature. Therefore, a sliding window of historical poses must be maintained in the estimator state.

B. Computational Challenges

As seen from the preceding section, forming multi-view geometric constraints from feature tracks across frames requires maintaining not only the current IMU state but also a set of cloned historical poses. As a result, conventional VIO systems maintain a large size of states primarily to support multi-frame visual constraints. This could entail hostile computational requirements for resource-limited platforms.

To see this, let n denote the estimator state dimension and m the measurement dimension. In MSCKF-like VIO, the measurement update is dominated by matrix operations in the Kalman gain and covariance update, which scale as $\mathcal{O}(mn^2)$ when $m > n$. Since the state dimension grows with the number of cloned poses and maintained feature tracks, the update cost increases quadratically with the sliding window size, leading to a significant computational bottleneck.

This precisely motivates our new minimalist formulation as shown in Section IV, in which we maintain only a minimal set of navigation states, making the propagation and update require constant complexity, and the inferred motion constraint scales linearly to measurement size as $\mathcal{O}(m)$. All this becomes possible is only due to the novel inferred motion constraint that is derived next.

III. Inferred Motion Constraints

As the key enabler for the proposed miniVIO, we derive an inferred motion measurement that constrains only the current navigation states, while extracting as much information as possible from the multi-view feature tracks. Note that multi-view feature geometry provides only the *direction* of this relative translation, whereas IMU preintegration provides same translation through inertial kinematics with metric scale. With this key observation, through the inertial preintegration, we can relate the translational direction induced from visual features to the navigation states, thus constructing the measurement model, without features and clones involved.

Specifically, we consider two keyframes $\{C_{k-1}\}$ and $\{C_k\}$ with corresponding IMU frames $\{I_{k-1}\}$ and $\{I_k\}$ at timestamps t_{k-1} and t_k, as illustrated in Fig. 2. This two-frame example is used only to simplify the presentation of the derivation. In practice, constraints can be chosen at any keyframe $\{I_a\}$. Moreover, it can be constructed from multiple keyframes and combined together within the same formulation. Expressing the relative motion with respect to

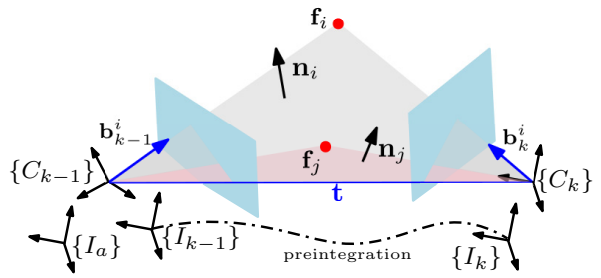

Fig. 2: Geometric illustration of the inferred motion constraint between two frames. The relative motion between frames $k-1$ and k is obtained from IMU preintegration, while visual feature observations provide directional constraints. $\mathbf{f}_i$ and $\mathbf{f}_j$ are feature points, the bearing vectors $\mathbf{b}^i_{k-1}$ and $\mathbf{b}^i_k$ define viewing rays toward $\mathbf{f}_i$, whose surface normals $\mathbf{n}_i$ determine the constraint direction $\mathbf{t}$. I_a is the reference frame.

this reference frame allows constraints to be constructed from arbitrary sets of keyframes and enables multiple constraints over different time intervals to be written in a unified form. From this design, we remove the need to maintain stochastic pose clones in the estimator state, since all constraints can be expressed directly as functions of the reference-frame motion variables. We then exploit the fact that the relative translation between the two frames can be expressed using inertial kinematics as a function of the reference frame velocity and gravity. Combining this with the translation direction from multi-view visual observations, we obtain a linear constraint that relates the relative velocity and gravity as:

$$\mathbf{A}\,\mathbf{x}_{I_a} - \mathbf{b} = \mathbf{0} \tag{8}$$

where the relative state is defined as local velocity and gravity in the reference frame: $\mathbf{x}_{I_a} = \begin{bmatrix} {}^{I_a}\mathbf{v}_{I_a}^\top & {}^{I_a}\mathbf{g}^\top \end{bmatrix}^\top$. In the following, we will present detailed derivation on how to derive this inferred motion constraint.

A. IMU: Metric Relative Translation

We briefly summarize the inertial model required for the induced measurement derivation. A standard 6-axis IMU provides local linear acceleration and angular velocity measurements $\mathbf{a}_m$ and $\boldsymbol{\omega}_m$. Following standard IMU preintegration theory [6], [7], [20], the relative motion between the reference frame $\{I_a\}$ and any later IMU keyframe $\{I_\kappa\}$ can be obtained by integrating IMU measurements over the interval $[t_a, t_\kappa]$:

$$ {}^{I_\kappa}_{I_a}\mathbf{R} := {}^{I_\kappa}_{I_a}\Delta\mathbf{R} \tag{9}$$

$$ {}^{I_a}\mathbf{p}_{I_\kappa} = {}^{I_a}\mathbf{v}_{I_a}\Delta T_\kappa + \frac{1}{2}\,{}^{I_a}\mathbf{g}\,\Delta T_\kappa^2 + {}^{I_a}\boldsymbol{\alpha}_\kappa \tag{10}$$

$$ {}^{I_a}\mathbf{v}_{I_\kappa} = {}^{I_a}\mathbf{v}_{I_a} + {}^{I_a}\mathbf{g}\,\Delta T_\kappa + {}^{I_a}\boldsymbol{\beta}_\kappa \tag{11}$$

where $\Delta T_\kappa = (t_\kappa - t_a)$ is the time span for integration, ${}^{\kappa}_{I_a}\Delta\mathbf{R}$, ${}^{I_a}\boldsymbol{\alpha}_\kappa$ and ${}^{I_a}\boldsymbol{\beta}_\kappa$ are obtained by IMU preintegration [7]. In the following, we derive the relative translation between two keyframes as illustrated in Figure 2. The relative translation ${}^{I_a}\Delta\mathbf{p}$ can be derived as:

$$ {}^{I_a}\Delta\mathbf{p} = {}^{I_a}\mathbf{p}_{I_\kappa} - {}^{I_a}\mathbf{p}_{I_{\kappa-1}} \tag{12}$$

where ${}^{I_a}\mathbf{p}_{I_\kappa}$ and ${}^{I_a}\mathbf{p}_{I_{\kappa-1}}$ can be obtained from the IMU measurements via position preintegration (see (10)). Substituting the ${}^{I_a}\mathbf{p}_{I_\kappa}$ and ${}^{I_a}\mathbf{p}_{I_{\kappa-1}}$ computed by (10) into (12), we get a linear relationship:

$$ {}^{I_a}\Delta\mathbf{p} = \mathbf{A}'\,\mathbf{x}_{I_a} + \mathbf{b}' \tag{13}$$

where $\mathbf{A}' = \begin{bmatrix} (\Delta T_k - \Delta T_{k-1})\,\mathbf{I}_3 & \frac{1}{2}\left(\Delta T_k^2 - \Delta T_{k-1}^2\right)\mathbf{I}_3 \end{bmatrix}$, $\mathbf{b}' = {}^{I_a}\boldsymbol{\alpha}_k - {}^{I_a}\boldsymbol{\alpha}_{k-1}$. Eq. (13) provide an explicit expression for the relative translation between the two keyframes in terms of inertial preintegration quantities, which can be written as a linear function of the reference-frame velocity and gravity.

B. Vision: Translational Direction

We seek to recover the direction of the relative translation between two keyframes using visual observations alone. While monocular geometry cannot determine the translation magnitude, it fully constrains its direction. We denote this scale-free direction by $\mathbf{t}$, the relative translation thus satisfies ${}^{I_a}\Delta\mathbf{p} = s\,\mathbf{t}$ with unknown scale $\mathbf{s}$. As shown in Fig. 2, we illustrate the geometry using two example features observed in two keyframes. Let $\mathbf{b}^l_\kappa$ denote the normalized bearing from keyframe $\kappa \in \{k-1, k\}$ to feature l. For each feature observed in both keyframes, the two bearing rays define an epipolar plane together with the relative translation direction. Since the translation direction lies in this plane, it must be orthogonal to the plane normal. Expressed in the common reference frame $\{I_a\}$, the normal of the epipolar plane associated with feature $\mathbf{f}_i$ is

$$\mathbf{n}_i = {}^{I_a}\mathbf{b}^i_{k-1} \times {}^{I_a}\mathbf{b}^i_k \tag{14}$$

Here the bearing of feature $\mathbf{f}_i$ observed in keyframe $\kappa \in \{k-1, k\}$ is rotated into $\{I_a\}$ as ${}^{I_a}\mathbf{b}^i_\kappa = {}^{I_a}_{I_\kappa}\mathbf{R}\,{}^{I}_{C}\mathbf{R}\,\mathbf{b}^i_\kappa$, where ${}^{I_a}_{I_\kappa}\mathbf{R}$ is obtained from IMU preintegration (9), and ${}^{I}_{C}\mathbf{R}$ denotes the known camera–IMU extrinsic orientation. Accordingly, for each feature observed in both keyframes, the coplanarity condition yields the orthogonality constraint $\mathbf{n}_i^\top\mathbf{t} = 0, \quad i = 1, \ldots, N_f$ where $\mathbf{n}_i$ is the epipolar-plane normal computed from the corresponding bearing pair. Summing these constraints gives

$$\mathbf{M}\mathbf{t} = \mathbf{0}, \quad \mathbf{M} := \sum_{i=1}^{N_f} \mathbf{n}_i\mathbf{n}_i^\top \tag{15}$$

The matrix $\mathbf{M}$ is symmetric and positive semi-definite by construction. In the ideal noise-free case, the normals span a two-dimensional subspace orthogonal to the true translation direction. Consequently, $\mathbf{M}$ has rank two and admits a one-dimensional nullspace spanned by $\mathbf{t}$. Under noisy measurements, $\mathbf{M}$ becomes full rank. Nevertheless, the translation direction corresponds to the eigenvector associated with the smallest eigenvalue of $\mathbf{M}$. That is, if $\mathbf{M} = \mathbf{E}'\boldsymbol{\Lambda}\mathbf{E}'^\top$, where $\mathbf{E}' = [\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3]$ with eigenvalues ordered as $0 \leqslant \lambda_1 \leqslant \lambda_2 \leqslant \lambda_3$, then the translation direction $\mathbf{t}$ lies along $\mathbf{e}_1$. The remaining eigenvectors span the subspace orthogonal to $\mathbf{t}$. For convenience, we define $\mathbf{E} := [\mathbf{e}_2, \mathbf{e}_3]$, which will be used in the subsequent formulation.

C. Multi-View Constraint Reformulation

Having obtained two complementary representations of the same relative translation, we now derive a constraint on the reference state. Since both describe the same physical motion, the inertial-predicted translation must be collinear with the vision-estimated direction $\mathbf{t}$. This implies that the inertial translation has no component orthogonal to $\mathbf{t}$.

$$\mathbf{A}'\mathbf{x}_{I_a} + \mathbf{b}' = s\,\mathbf{t} \tag{16}$$

This collinearity condition implies that the inertial-predicted translation must have no component orthogonal to the direction $\mathbf{t}$. In other words, its projection onto the subspace

perpendicular to $\mathbf{t}$ vanishes and results a linear constraint on the reference state:

$$\mathbf{E}^\top(\mathbf{A}'\,\mathbf{x}_{I_a} + \mathbf{b}') = \mathbf{0} \quad \Rightarrow \quad \mathbf{A}\mathbf{x}_{I_a} = \mathbf{b} \tag{17}$$

where $\mathbf{A} = \mathbf{E}^\top\mathbf{A}'$ and $\mathbf{b} = -\mathbf{E}^\top\mathbf{b}'$.

Remark 1. *In practice, the proposed constraint is not limited to a single keyframe pair. Because all quantities are expressed with respect to the common reference frame $\{I_a\}$, constraints can be constructed from arbitrary historical keyframes that have sufficient visual overlap with the current frame and combined within the same formulation. Importantly, the inferred constraint can ultimately be reparameterized as a function of the current navigation state. Importantly, the proposed formulation does not require velocity and gravity to be explicitly recoverable. Instead, they serve as intermediate variables used to construct the inferred constraint, which can subsequently be mapped to the current navigation state for state update.*

IV. Minimalist VIO

We now incorporate the inferred motion constraint derived in the previous section into a new minimal-state VIO framework, termed **miniVIO**, as summarized in Algorithm 1. Specifically, our state vector just contains the bare minimum 9DoF navigation states (position, rotation and velocity)—to be *minimalist*:

$$\mathbf{x}_{I_k} = \begin{bmatrix} {}^{I_k}_{G}\bar{q}^\top & {}^{G}\mathbf{p}_{I_k}^\top & {}^{G}\mathbf{v}_{I_k}^\top \end{bmatrix}^\top \tag{18}$$

which is propagated forward based on the IMU kinematics (4) as in MSCKF-based VIO [19].

Remark 2. *Compared with conventional VIO formulations (18), our state does not explicitly include IMU biases that are unobservable given our inferred motion constraints. While prior work has well established the importance of IMU bias estimation, particularly gyroscope bias estimation, in VINS, our design choice is motivated by several practical observations. [21]: First, IMU biases typically evolve slowly over time and may not need to estimate at the same rate as the navigation state. Second, bias estimates can be reliably recovered when additional information sources become available, such as visual constraints or system reinitialization. Third, IMU biases can also be estimated using a separate lightweight AHRS (3DoF) filter without increasing the dimension of the main estimator state. Nevertheless, it would incur marginal overhead to include biases if needed.*

A. Update with Inferred Motion Constraints

We now show how the inferred constraint can be formulated as a measurement function that directly updates the navigation state. Whenever a valid inferred constraint is constructed (17), it can be written as a compact linear constraint on the reference state $\mathbf{x}_{I_a}$:

$$\mathbf{r}_a(\mathbf{x}_{I_a}) \triangleq \mathbf{A}\mathbf{x}_{I_a} - \mathbf{b} = \mathbf{0} \tag{19}$$

However, the estimator is parameterized at the current time t_k with state $\mathbf{x}_{I_k}$. We re-parameterize the constraint in terms of the current state so that it can be incorporated as a single-time measurement update. We now derive the nonlinear mapping function $\mathbf{g}(\mathbf{x}_k)$ that relates the reference state $\mathbf{x}_{I_a}$ to the current estimator state $\mathbf{x}_{I_k}$. From the IMU preintegration relation over the interval $[t_a, t_k]$ (see (11)), the velocity expressed in the reference frame satisfies:

$${}^{I_a}\mathbf{v}_{I_a} = {}^{I_a}\mathbf{v}_{I_k} - {}^{I_a}\mathbf{g}\,\Delta T_k - {}^{I_a}\boldsymbol{\beta}_k \tag{20}$$

Algorithm 1 miniVIO

Propagation:
- Propagate the state and sqaure-root covariance to time k.

Inferred Motion Constraint Formulation:
- Select keyframes from the feature tracking database.
- Calculate preintegration terms between keyframes (9) (10) (11).
- Calculate the nullspace of the translation direction across any two keyframe pairs (14) (15).
- Combine the nullspace with preintegration to formulate constraint with respect to velocity and gravity direction (13) (16) (17).

Update with Inferred Motion Constraints
- Map the constraint to current state $\mathbf{x}_{I_k}$ (23), then calculate residual and Jacobian (24) (25) and update the state and sqaure-root covariance with SRF update.

The quantities ${}^{I_a}\mathbf{v}_{I_k}$ and ${}^{I_a}\mathbf{g}$ can be obtained from the current state through frame transformations. Specifically,

$${}^{I_a}\mathbf{v}_{I_k} = {}^{I_a}_{I_k}\mathbf{R}\,{}^{I_k}_{G}\mathbf{R}\,{}^{G}\mathbf{v}_{I_k}, \quad {}^{I_a}\mathbf{g} = {}^{I_a}_{I_k}\mathbf{R}\,{}^{I_k}_{G}\mathbf{R}\,{}^{G}\mathbf{g} \tag{21}$$

where ${}^{I_k}_{G}\mathbf{R}$ is the orientation and ${}^{I_a}_{I_k}\mathbf{R}$ is obtained from the preintegrated rotation. Substituting these relations into (20) gives the reference-frame velocity as a function of the current state. Stacking the expressions for ${}^{I_a}\mathbf{v}_{I_a}$ and ${}^{I_a}\mathbf{g}$ yields the nonlinear mapping:

$$\mathbf{x}_{I_a} = \mathbf{h}(\mathbf{x}_{I_k}) \tag{22}$$

$$= \begin{bmatrix} {}^{I_a}_{I_k}\mathbf{R}\,{}^{I_k}_{G}\mathbf{R}\,{}^{G}\mathbf{v}_{I_k} - \left({}^{I_a}_{I_k}\mathbf{R}\,{}^{I_k}_{G}\mathbf{R}\,{}^{G}\mathbf{g}\right)\Delta T_k - {}^{I_a}\boldsymbol{\beta}_k \\ {}^{I_a}_{I_k}\mathbf{R}\,{}^{I_k}_{G}\mathbf{R}\,{}^{G}\mathbf{g} \end{bmatrix} \tag{23}$$

Using this mapping, the inferred constraint can be evaluated directly in terms of the current state. Substituting into (19) gives an equivalent constraint on $\mathbf{x}_{I_k}$:

$$\mathbf{r}_k \triangleq \mathbf{A}\mathbf{h}(\mathbf{x}_{I_k}) - \mathbf{b} = \mathbf{0} \tag{24}$$

which can be linearized as

$$\mathbf{r}_k \simeq \mathbf{H}_k\tilde{\mathbf{x}}_{I_k} + \mathbf{n}_k, \quad \mathbf{H}_k = \frac{\partial \mathbf{r}_k}{\partial \mathbf{x}_{I_k}} = \mathbf{A}\frac{\partial \mathbf{g}}{\partial \mathbf{x}_{I_k}} \tag{25}$$

Here, $\mathbf{n}_k$ denotes the noise that models uncertainty in the inferred measurement. The resulting system can then be incorporated into the estimator using a standard EKF or square-root filtering update. Note that the noise for of the inferred measurement depends on tracking accuracy, tracking outliers, preintegration error, etc. In this work, we approximate this uncertainty using a Gaussian covariance that is empirically tuned. A more rigorous uncertainty quantification method is left for future work. We should note that although the measurement does not directly constrain position, the observable component of it can still be updated through its correlation with velocity.

B. Summary and Discussion

The proposed formulation eliminates redundant auxiliary states, such as stochastic pose clones and 3D environmental landmarks, retaining only the bare essential motion variables required by downstream applications. As a result, the dimensions of the proposed estimator are significantly lower than those of conventional VIO systems. Although MSCKF-like VIO can also reduce the state size by maintaining a smaller number of cloned poses, its accuracy can degrade significantly because the triangulated 3D points tend to be of relatively poor quality. In contrast, our proposed method does not suffer from this issue, as it does not require 3D point triangulation. To update this minimal state without introducing additional auxiliary variables, we derive the

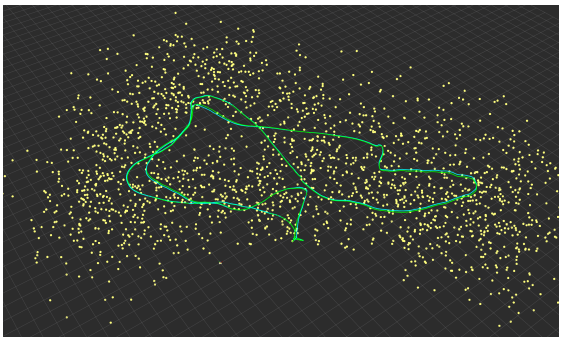

Fig. 3: EurocMAV Machine Hall 05 as simulated trajectory.

inferred constraints on velocity and gravity for the following reasons:

- Both velocity and gravity direction are foundational to VIO; their accurate estimation is a prerequisite for successful system initialization and bootstrapping.
- Velocity is a critical yet fragile component of the state. Unbounded velocity drift leads to significant pose inaccuracies, which in turn causes 3D point triangulation to fail. Without the visual constraints provided by traditional reprojection errors, this drift can escalate, leading to catastrophic system failure. Furthermore, unlike the gravity vector—which can be maintained relatively well by modern MEMS IMU—velocity estimation relies solely on accelerometer integration. This process is inherently coupled with gravity; any error in orientation leads to a miscalculation of the gravity component. For instance, a small 1° orientation error introduces approximately 0.17 m/s^2 of acceleration error, which accumulates rapidly during integration.
- We include gravity as a constraint to ensure its long-term stability within our formulation. If gravity were assumed to be a fixed "true" value, we could derive a constraint solely for velocity, but the gravity direction itself would remain unconstrained. While a standalone 3-DoF filter could theoretically bound gravity drift using accelerometer readings, such filters are often contaminated by external (non-gravitational) accelerations. Our formulation avoids this contamination, providing a cleaner update for the gravity vector.

It is important to note that, unlike methods that rely on explicit velocity and gravity updates, our approach does not require these parameters to be fully recoverable. Consequently, even during degenerate motion, the system can still provide constraints for the observable directions.

Remark 3. *Compared to methods that rely directly on translation direction for update (e.g., [16]), our minimalist formulation offers two primary advantages:*

- *Our approach eliminates the need to maintain stochastic clones within the state vector. Consequently, constraints can be formulated using any historical frame, provided there is sufficient visual overlap with the current frame.*
- *Given the critical role of velocity in system stability, our formulation establishes a direct constraint on velocity rather than on position. This avoids the explicit reliance on the correlation between position and velocity, which might be inaccurate in practice.*

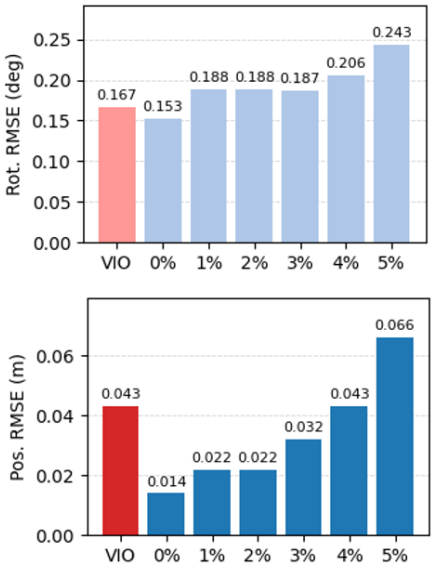

Fig. 4: Pose RMSE in simulation with relative velocity measurement under different-level of noise. We apply Gaussian noise to each axis of the relative velocity measurement, and the σ for the noies is the norm of the velocity multiple the noise percentage. The red bar corresponds to baseline $\sqrt{\text{VINS}}$.

V. Numerical Study

To evaluate the significance of velocity in VINS, we developed an estimator utilizing a navigation state. In this setup, the IMU performs state propagation, while a local velocity measurement ${}^I\mathbf{v}_I$ is leveraged for the state update. These velocity measurements are derived from B-spline differentiation from simulated trajectories, with varying levels of Gaussian noise added to assess their impact on estimation accuracy. Specifically, we scale the noise standard deviation, σ, relative to the norm of the current velocity. As a baseline for comparison, we use the state-of-the-art filtering-based system $\sqrt{\text{VINS}}$, which follows the MSCKF-VIO formulation (see Section II-A). Simulation data was generated using the OpenVINS simulator [19], which provides realistic IMU readings and image-bearing measurements given a real-world robot trajectory. Simulation parameters and the resulting trajectory are detailed in Table I and Fig. 3, respectively.

The results illustrated in Fig. 4 reveal a compelling finding: given the current simulation configuration, the "minimalist" system achieves accuracy comparable to, or in some cases exceeding, full VIO performance when provided with ground truth or low-noise velocity measurements. While accuracy degrades as measurement noise increases, the error remains within a functionally useful range rather than resulting in catastrophic failure. This experiment validates our underlying assumption and confirms the viability of our proposed minimal state formulation.

VI. Real World Experiment

We built our system on top of recent $\sqrt{\text{VINS}}$ [22], [23], a square-root filter-based VINS system, hightlighted by its efficiency. We evaluated our system on 3 real-world datasets, an MAV dataset EurocMAV [24], one handheld dataset TUM-VI [25], and one ego-centric Aria Every Day(AEA)

TABLE I: Simulation parameters and setup for baseline VIO.

Parameter	Value	Parameter	Value
Gyro. White Noise	2.0e-4	Gyro. Rand. Walk	2.0e-5
Accel. White Noise	2.0e-3	Accel. Rand. Walk	3.0e-4
Cam Freq. (Hz)	10	IMU Freq. (Hz)	400
Num. Clones	11	Tracked Feat.	100
Max. MSCKF Feat.	40	Sequence Length (m)	91.5

activities dataset [26]. In EurocMAV and TUM-VI, only left camera is used, while in AEA dataset, both the 2 cameras are used without stereo constraints as they have minimal overlap. miniVIO keeps minimal state, and uses a maximal of 0.5-seconds window, and 3 keyframes to formulate inferred constraints. The noise σ in EurocMAV and TUM-VI is set as 0.02, while in AEA, we set it as 0.05 across 140 sequences. To ensure good initial state and covariance, we use $\sqrt{\text{VINS}}$ to bootstrap the system for 10 seconds, then all the states other than navigation state are marginalized along with their corresponding covariance. Biases are also fixed after this point. miniVINS shares the same visual front-end as $\sqrt{\text{VINS}}$, where 200 features are tracked with KLT. we compare miniVIO against two representative visual–inertial odometry systems: VINS-Mono [2] and the square-root filtering framework $\sqrt{\text{VINS}}$ [22]. VINS-Mono represents a widely used sliding-window optimization-based VIO system, while $\sqrt{\text{VINS}}$ is a recent filtering-based formulation designed for computational efficiency. We use both default configs for VINS-Mono (150 features) and $\sqrt{\text{VINS}}$ (200 features, 11 clones, 50 SLAM features, 40 MSCKF features) as in their public repo.

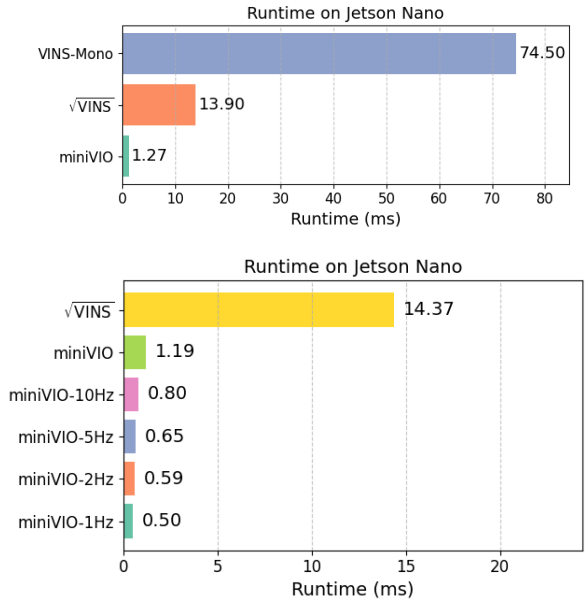

Fig. 5: Runtime comparison on the Jetson Nano platform. **Top:** EuRoC MAV dataset. **Bottom:** TUM-VI dataset. The proposed miniVIO significantly reduces the per-frame runtime compared with $\sqrt{\text{VINS}}$. We additionally evaluate different velocity measurement frequencies (10 Hz–1 Hz), demonstrating that miniVIO remains computationally lightweight even at high update rates.

A. Algorithm Efficiency

We evaluate the computational efficiency of the proposed miniVIO algorithm on a Jetson Nano platform equipped with an ARM Cortex-A57 CPU (2014) running at 1.5 GHz.

Figure 5 (top) reports the average runtime per frame on the EuRoC MAV dataset. VINS-Mono requires approximately **74.5 ms** per frame, while $\sqrt{\text{VINS}}$ reduces the runtime to **13.9 ms** through an efficient square-root filtering formulation. In contrast, miniVIO requires only **1.27 ms** per frame. This corresponds to more than **10**× **speedup** compared with $\sqrt{\text{VINS}}$ and nearly **60**× **speedup** compared with VINS-Mono. To further understand the runtime characteristics, Figure 1 presents the runtime composition of miniVIO. The total runtime per frame is **1.27 ms**, which consists of four components: state propagation, database management, inferred constraint generation, and measurement update. Among these modules, inferred constraint generation accounts for **55.1%** of the runtime, followed by propagation (**24.0%**), database management (**13.5%**), and measurement update (**7.4%**). Despite being the largest component, inferred constraint generation remains lightweight in absolute terms. Moreover, the current implementation serves as a research prototype and has not yet been fully optimized for real-time engineering constraints. For instance, the system currently performs a full re-integration of IMU measurements for each update; implementing incremental pre-integration and caching historical results would significantly reduce the computational overhead.

Fig. 5 (bottom) further shows the runtime comparison on the TUM-VI dataset. The baseline $\sqrt{\text{VINS}}$ requires **14.37 ms** per frame, whereas the proposed miniVIO requires only **1.19 ms**, achieving more than an order-of-magnitude speedup. We further evaluate the runtime under different velocity measurement frequencies. As shown in Fig. 5, reducing the update frequency from 10 Hz to 1 Hz decreases the runtime from **0.80 ms** to **0.50 ms** per frame. Consequently, as the update frequency decreases, the computational overhead of the induced constraints is further reduced. In the limit, the total processing time converges to the baseline required for state propagation and database management, both of which are computationally negligible.

B. Accuracy Evaluation

We report average absolute trajectory error (ATE) as the primary metric because miniVIO targets 6-DoF trajectory estimation, and ATE directly reflects the accumulated pose accuracy of the estimated trajectory. We evaluate the accuracy of miniVIO on the EuRoC MAV dataset [24]. Table II reports the ATE in degrees and meters across all sequences. Overall, miniVIO achieves performance that is comparable to state-of-the-art VIO systems such as $\sqrt{\text{VINS}}$ and VINS-Mono. Despite relying on the inferred constraint instead of explicit feature reprojection factors, the estimator maintains similar levels of accuracy across both the Vicon Room and Machine Hall sequences.

We further evaluate miniVIO on the Aria Everyday Activities (AEA) dataset [26], where a total of 140 sequences (maximum ∼17 min) in 5 different locations and across 7 devices have been tested. Table III reports the ATE results across five recording locations. Compared with EuRoC, the AEA dataset introduces additional challenges, including inaccurate

TABLE II: Average Absolute Trajectory Error (ATE) in degrees/meters in EurocMAV.

Algo.	V101	V102	V103	V201	V202	V203	MH01	MH02	MH03	MH04	MH05
miniVIO	0.74 / 0.08	3.67 / 0.13	2.73 / 0.16	1.84 / 0.24	1.66 / 0.11	1.03 / 0.24	1.44 / 0.19	1.25 / 0.33	2.62 / 0.42	1.55 / 0.46	2.19 / 0.39
$\sqrt{\text{VINS}}$	0.54 / 0.06	1.65 / 0.05	2.68 / 0.06	1.09 / 0.10	1.48 / 0.06	1.17 / 0.11	1.97 / 0.10	0.74 / 0.14	0.88 / 0.11	0.99 / 0.25	1.13 / 0.35
VINS-Mono	0.82 / 0.07	2.74 / 0.10	5.15 / 0.15	2.13 / 0.09	2.57 / 0.13	3.43 / 0.29	0.78 / 0.20	0.86 / 0.18	1.82 / 0.23	2.51 / 0.41	0.94 / 0.29

TABLE III: ATE Results on the Aria Everyday Activities dataset. As AEA dataset contains certain challenges: inaccurate camera model, dynamic objects, textureless environment, some of the sequences yields significant larger error, which degrades the overall average. To better demonstrate the performance, we set several position errors as thresholds and only average the runs with errors smaller than the thresholds and the success rate (e.g. **miniVIO**(@0.5m) means using a 0.5 meter position error as threshold).

Algo.	Location 1	Location 2	Location 3	Location 4	Location 5	Avg.	Success Rate
$\sqrt{\text{VINS}}$	1.13 / 0.04	0.95 / 0.04	1.13 / 0.04	1.42 / 0.05	1.15 / 0.06	1.16 / 0.05	100%
miniVIO	3.48 / 0.38	2.63 / 0.31	2.37 / 0.30	1.85 / 0.21	3.00 / 0.44	2.66 / 0.32	100%
miniVIO(@1m)	3.40 / 0.23	2.13 / 0.25	2.37 / 0.30	1.85 / 0.21	2.95 / 0.38	2.45 / 0.26	97.9%
miniVIO(@0.5m)	3.32 / 0.20	1.93 / 0.19	1.92 / 0.21	1.79 / 0.19	2.74 / 0.34	2.26 / 0.21	85.0%
miniVIO(@0.25m)	2.06 / 0.16	1.74 / 0.14	1.64 / 0.16	1.70 / 0.14	2.69 / 0.21	1.81 / 0.15	61.4%

TABLE IV: Average Absolute Trajectory Error (ATE) in degrees/meters in TUM-VI varying update frequency.

Algo.	room1_512_16	room2_512_16	room3_512_16	room4_512_16	room5_512_16	room6_512_16	Average
miniVIO(20Hz)	1.78 / 0.09	1.22 / 0.12	2.80 / 0.12	1.45 / 0.14	4.09 / 0.15	0.93 / 0.07	2.05 / 0.11
miniVIO(10Hz)	1.55 / 0.09	1.32 / 0.12	2.69 / 0.13	1.43 / 0.15	3.24 / 0.14	1.01 / 0.07	1.87 / 0.12
miniVIO(5Hz)	1.51 / 0.09	1.29 / 0.12	2.71 / 0.13	1.28 / 0.16	2.32 / 0.13	1.01 / 0.08	1.69 / 0.12
miniVIO(2Hz)	1.51 / 0.10	1.40 / 0.09	2.41 / 0.21	1.29 / 0.17	1.65 / 0.17	0.91 / 0.08	1.53 / 0.14
miniVIO(1Hz)	1.61 / 0.15	1.18 / 0.09	2.29 / 0.72	1.37 / 0.24	2.73 / 0.28	0.88 / 0.14	1.68 / 0.27
$\sqrt{\text{VINS}}$(20Hz)	1.00 / 0.09	1.15 / 0.14	1.03 / 0.06	1.20 / 0.04	1.11 / 0.11	1.08 / 0.07	1.09 / 0.08

camera models, dynamic objects, large textureless regions and dynamic motions (such as users performing exercises like push-ups and sit-ups). As a result, some sequences produce significantly larger errors, which degrade the overall average performance. To better analyze the estimator behavior, we additionally report results under several position-error thresholds. Specifically, we compute the average error only over runs whose final position error is below a given threshold, and report the corresponding success rate (e.g., miniVIO (@0.5m) denotes using a 0.5 m position-error threshold). As shown in Table III, while conventional $\sqrt{\text{VINS}}$ achieves lower average error on successful runs, miniVIO maintains competitive performance while achieving high success rates under practical thresholds. The slightly higher errors observed in some sequences may also be attributed to the absence of explicit SLAM landmarks. These results indicate that mini can maintain reliable motion estimation even in challenging real-world conditions.

C. Robustness to Update Interval

We further evaluate the robustness of the proposed inferred measurement under varying update intervals using the TUM-VI dataset, where a 20 Hz camera is available. In this experiment, the inferred constraint is applied only once given different frequency, simulating scenarios where visual updates are sparse or computational resources are limited.

Table IV reports the average absolute trajectory error (ATE) under different update intervals ranging from $0\,\text{s}$ (every frame) to $1.0\,\text{s}$. As the update interval increases, the estimator receives fewer visual constraints, which naturally leads to a increase in trajectory error. Nevertheless, miniVIO maintains stable performance even when the update frequency is reduced. For example, when the update interval increases to $0.5\,\text{s}$, miniVIO still achieves an average error of $1.53^\circ/0.14\,\text{m}$, indicating that the inferred constraint remains informative even at low update rates. Even at $1.0\,\text{s}$ intervals, the estimator continues to produce reasonable trajectory estimates. This robustness is likely due to the fact that the inferred constraint captures geometric motion cues derived from multi-view observations, providing informative updates even when the measurement frequency is reduced. As a result, each update carries relatively rich motion information, allowing the estimator to operate reliably even when updates are applied less frequently.

This property is particularly beneficial for resource- constrained or event-driven systems, where visual measurements may be processed intermittently due to limited computational budget or power constraints. Another advantage of the proposed formulation is that the reference state does not need to be explicitly estimated. Instead, the inferred constraint is directly formulated as a measurement on the estimator state. This avoids solving intermediate variables that may become ill-conditioned in degenerate configurations, improving the numerical robustness of the measurement construction.

VII. Discussion and Limitation

While our miniVIO achieves impressive performance, a few limitations should be noted:

- The uncertainty of the inferred measurement is not yet analytically quantified. Accurately modeling this uncertainty is challenging for several reasons. The inferred constraint is constructed from multiple intermediate quantities, including feature bearings, epipolar geometry, and IMU preintegration, each of which introduces its own uncertainty. Moreover, the derivation involves several simplifying assumptions and reuses information that is also involved in the propagation process, which introduces nontrivial correlations between different er-

ror sources. In this work, the measurement noise covariance is therefore determined empirically, although we observe that a consistent parameter performs reliably across datasets given fixed keyframe and window configurations. Rigorously quantifying this uncertainty remains a key objective for future work.

- Our miniVIO is more sensitive to certain initial parameters, such as IMU biases and camera calibrations, since these quantities are treated as known constants rather than being estimated online. Consequently, inaccurate initial estimates can distort the resulting velocity and gravity constraints. In practice, this sensitivity can be mitigated through standard VIO initialization procedures or by employing decoupled estimators (e.g., a standalone 3-DoF filter) to provide real-time updates for critical states like biases.

VIII. Conclusions and Future Work

In this work, we presented miniVIO, a minimalist visual– inertial odometry framework that maintains a 9 DoF estimator state. Unlike conventional VIO systems that rely on state augmentation to incorporate multi-view geometric information, the proposed approach reformulates feature tracks and inertial dynamics into a compact inferred inferred motion constraints, which directly update the navigation state. Specifically, the constraint is formulated by requiring the IMU predicted relative translation to be collinear with the direction derived from multi-view visual observations. This relationship infer a constraint on the relative velocity and local gravity, which is then mapped to the current navigation state and used for efficient state update. This formulation allows visual information to be incorporated without introducing additional pose clones or landmark variables, enabling visual constraints to directly update the navigation state while maintaining a fixed estimator size. As a result, the proposed estimator achieves minimal update complexity while still preserving informative motion constraints. Extensive experiments demonstrate that miniVIO achieves competitive accuracy compared with state-of-the-art VIO systems while significantly reducing computational overhead, enabling efficient operation on resource-constrained platforms. This formulation also enables asynchronous and low-frequency visual updates, since the inferred constraint can be constructed and applied independently of the camera frame rate, this can further save energy.

Future work will investigate uncertainty quantification of inferred constraints and event-triggered updates that allow visual constraints to be applied only when necessary. We are also interested to extend miniVIO design to VI-SLAM to incorporate loop closures.

References

[1] G. Huang, "Visual-inertial navigation: A concise review," in *Proc. International Conference on Robotics and Automation*, Montreal, Canada, May 2019.

[2] 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.

[3] V. Usenko, N. Demmel, D. Schubert, J. Stückler, and D. Cremers, "Visual-inertial mapping with non-linear factor recovery," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 422–429, 2019.

[4] C. Campos, R. Elvira, J. J. G. Rodríguez, J. M. Montiel, and J. D. Tardós, "Orb-slam3: An accurate open-source library for visual, visual–inertial, and multimap slam," *IEEE Transactions on Robotics*, vol. 37, no. 6, pp. 1874–1890, 2021.

[5] C. Chen, P. Geneva, Y. Peng, W. Lee, and G. Huang, "Optimization-based vins: Consistency, marginalization, and fej," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2023.

[6] C. Forster, L. Carlone, F. Dellaert, and D. Scaramuzza, "On-manifold preintegration theory for fast and accurate visual-inertial navigation," *IEEE Transactions on Robotics*, pp. 1–18, 2015.

[7] K. Eckenhoff, P. Geneva, and G. Huang, "Closed-form preintegration methods for graph-based visual-inertial navigation," *International Journal of Robotics Research*, vol. 38, no. 5, pp. 563–586, 2019.

[8] Y. Yang, B. P. W. Babu, C. Chen, G. Huang, and L. Ren, "Analytic combined imu integrator for visual-inertial navigation," in *Proc. of the IEEE International Conference on Robotics and Automation*, Paris, France, 2020.

[9] M. Kaess, A. Ranganathan, and F. Dellaert, "isam: Incremental smoothing and mapping," *IEEE Transactions on Robotics*, vol. 24, no. 6, pp. 1365–1378, 2008.

[10] M. Kaess, H. Johannsson, R. Roberts, V. Ila, J. J. Leonard, and F. Dellaert, "isam2: Incremental smoothing and mapping using the bayes tree," *The International Journal of Robotics Research*, vol. 31, no. 2, pp. 216–235, 2012.

[11] V. Indelman, S. Williams, M. Kaess, and F. Dellaert, "Information fusion in navigation systems via factor graph based incremental smoothing," *Robotics and Autonomous Systems*, vol. 61, no. 8, pp. 721–738, 2013.

[12] S. Leutenegger, S. Lynen, M. Bosse, R. Siegwart, and P. Furgale, "Keyframe-based visual–inertial odometry using nonlinear optimization," *The International Journal of Robotics Research*, vol. 34, no. 3, pp. 314–334, 2015.

[13] K. Eckenhoff, L. Paull, and G. Huang, "Decoupled, consistent node removal and edge sparsification for graph-based SLAM," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems*, Daejeon, Korea, Oct. 2016, pp. 3275–3282.

[14] A. I. Mourikis and S. I. Roumeliotis, "A multi-state constraint kalman filter for vision-aided inertial navigation," in *Proceedings 2007 IEEE international conference on robotics and automation*. IEEE, 2007, pp. 3565–3572.

[15] Y. Yang, J. Maley, and G. Huang, "Null-space-based marginalization: Analysis and algorithm," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems*, Vancouver, Canada, Sept. 2017, pp. 6749–6755.

[16] S. Weiss, M. W. Achtelik, S. Lynen, M. Chli, and R. Siegwart, "Real-time onboard visual-inertial state estimation and self-calibration of mavs in unknown environments," in *2012 IEEE International Conference on Robotics and Automation*, 2012, pp. 957–964.

[17] D. Abeywardena, S. Huang, B. Barnes, G. Dissanayake, and S. Kodagoda, "Fast, on-board, model-aided visual-inertial odometry system for quadrotor micro aerial vehicles," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, 2016, pp. 1530–1537.

[18] X. Lu, Y. Zhou, J. Niu, S. Zhong, and S. Shen, "Event-based visual inertial velometer," in *Robotics: Science and Systems*, Delft, Netherlands, July 2024.

[19] P. Geneva, K. Eckenhoff, W. Lee, Y. Yang, and G. Huang, "Openvins: A research platform for visual-inertial estimation," in *Proc. of the IEEE International Conference on Robotics and Automation*, Paris, France, 2020. [Online]. Available: https://github.com/rpng/open_vins

[20] T. Lupton and S. Sukkarieh, "Visual-inertial-aided navigation for high-dynamic motion in built environments without initial conditions," *IEEE Transactions on Robotics*, vol. 28, no. 1, pp. 61 –76, Feb. 2012.

[21] J. Kaiser, A. Martinelli, F. Fontana, and D. Scaramuzza, "Simultaneous state initialization and gyroscope bias calibration in visual inertial aided navigation," *IEEE Robotics and Automation Letters*, vol. 2, no. 1, pp. 18–25, 2016.

[22] Y. Peng, C. Chen, K. Wu, and G. Huang, "sqrt-vins: Robust and ultra-fast square-root filter-based 3d motion tracking," *IEEE Transactions on Robotics*, Sept. 2025.

[23] Y. Peng, C. Chen, and G. Huang, "Ultrafast square-root filter-based VINS," in *Proc. International Conference on Robotics and Automation*, Yokohama, Japan, May 2024.

[24] M. Burri, J. Nikolic, P. Gohl, T. Schneider, J. Rehder, S. Omari, M. W. Achtelik, and R. Siegwart, "The euroc micro aerial vehicle datasets," *The International Journal of Robotics Research*, vol. 35, no. 10, pp. 1157–1163, 2016.

[25] D. Schubert, T. Goll, N. Demmel, V. Usenko, J. Stückler, and D. Cremers, "The tum vi benchmark for evaluating visual-inertial odometry," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 1680–1687.

[26] Z. Lv, N. Charron, P. Moulon, A. Gamino, C. Peng, C. Sweeney, E. Miller, H. Tang, J. Meissner, J. Dong, *et al.*, "Aria everyday activities dataset," *arXiv preprint arXiv:2402.13349*, 2024.