2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
Madrid, Spain, October 1-5, 2018

# Robocentric Visual-Inertial Odometry

Zheng Huai and Guoquan Huang

***Abstract*— In this paper, we propose a novel *robocentric* formulation of visual-inertial navigation systems (VINS) within a multi-state constraint Kalman filter (MSCKF) framework and develop an efficient, lightweight, *robocentric visual-inertial odometry* (R-VIO) algorithm for consistent localization in challenging environments using only monocular vision. The key idea of the proposed approach is to deliberately reformulate the 3D VINS with respect to a moving local frame (i.e., robocentric), rather than a fixed global frame of reference as in the standard world-centric VINS, and instead utilize high-accuracy relative motion estimates for global pose update. As an immediate advantage of using this robocentric formulation, the proposed R-VIO can start from an arbitrary pose, *without* the need to align its orientation with the global gravity vector. More importantly, we analytically show that the proposed robocentric EKF-based VINS does *not* undergo the observability mismatch issue as in the standard world-centric frameworks which was identified as the main cause of inconsistency of estimation. The proposed R-VIO is extensively tested through both Monte Carlo simulations and real-world experiments using different sensor platforms in different environments and shown to achieve competitive performance with the state-of-the-art VINS algorithms in terms of consistency, accuracy and efficiency.**

## I. Introduction and Related Work

Enabling high-precision 3D navigation on mobile devices and robots with minimal sensing of low-cost sensors holds potentially huge implications in the real applications ranging from mobile augmented reality to autonomous driving. To this end, inertial navigation offers a classical 3D localization solution which utilizes an inertial measurement unit (IMU) measuring the 3 degree-of-freedom (DOF) angular velocity and linear acceleration of the sensor platform on which it is rigidly attached. Typically, IMU works at a high frequency (e.g., 100$\sim$1000Hz) that enables it to sense highly dynamic motion; however, suffered from the corrupting sensor noise and bias, purely integrating IMU measurements can easily result in unusable motion estimates. This necessitates to fuse the aiding information from *at least* a single camera to limit the accumulated inertial navigation drifts, which results in the visual-inertial navigation systems (VINS).

In the past decade, we have witnessed significant progress on VINS, including both visual-inertial SLAM (VI-SLAM) and visual-inertial odometry (VIO), and many different algorithms have been developed (e.g., [1], [2], [3], [4], [5], [6], [7], [8] and references therein). However, almost all these algorithms are based on the concept of *world-centric* formulation – that is, to estimate the absolute motion which is modeled with respect to a fixed global frame of reference, such as the earth-centered earth-fixed (ECEF) or the north-east-down (NED) frame. In order to achieve accurate localization, such world-centric VINS usually require a particular initialization procedure to estimate the starting pose in the global frame of reference, which, however, is hard to guarantee the accuracy in some cases (e.g., quick start, or poor illumination). While an extended Kalman filter (EKF)-based world-centric VINS algorithm has the advantage of lower computational cost [1], [4] as compared to the batch optimization-based ones (which incur high computation due to performing relinearization [5], [6]), it may become *inconsistent*, primarily due to the fact that the EKF linearized systems have different observability properties from the corresponding underlying nonlinear systems [4], [9], [10]. The remedies for mitigating this issue include enforcing the correct observability [4], [10], [11] or employing an invariant error representation [12]. Therefore, one may ask: *Do we have to formulate VINS in the standard world-centric form?* The answer is *no*. Intuitively, inspired by how humans navigate – we might not remember the starting (global) pose after traveling a long distance while knowing well the relative motion within a recent, short time interval – we may relax the fixed global frame of VINS, instead, choose a moving local frame to better estimate relative motion which can be used later for global pose update.

Note that, this sensor-centered idea for localization can be traced back to 2D laser-based robocentric mapping [13], [14], where the global frame is treated as a feature which is observed from the moving robot frame and the relative motion measurements from an odometer are fused for pose update, while the composition step makes it possible to shift the local frame of reference during the motion. Following the similar idea, in [15] a camera-centered formulation shows the potential to fuse the visual information with the measurements from proprioceptive sensors (e.g., angular and linear velocity measurements). Both methods had been applied to the EKF-based SLAM while performing mapping in the local frames, thus limiting the global uncertainty and improving the consistency of estimation. It should also be noted that a robust VINS algorithm using a different robocentric formulation and sensor-fusion scheme was recently introduced [16]. In particular, its state vector includes both IMU states and the features which are reformulated with respect to the local IMU frame, while the camera measurements are fused with IMU in a *direct* fashion. Additionally, in contrast to [14], [15], it employs the *iterated* EKF to perform update without the composition step to shift local frame of reference.

In this paper, following the idea [14], [15], we reformulate the VINS problem with respect to a local IMU frame, while in contrast to [14], [15], [16] which keep features in the state vector and have to concern the increasing computational cost

This work was partially supported by the University of Delaware (UD) College of Engineering, UD Cybersecurity Initiative, NSF (IIS-1566129), DTRA (HDTRA1-16-1-0039), and Google.

The authors are with the Dept. of Mechanical Engineering, University of Delaware, Newark, DE 19716, USA {zhuai|ghuang}@udel.edu.

as more features are observed and included, we focus on an EKF-based visual-inertial odometry (EKF-VIO) framework, i.e., multi-state constraint Kalman filter (MSCKF) [1], whose stochastic cloning enables the VINS to process hundreds of features while only keep a small number of robot poses from which those features have been observed in the state vector, thus significantly reducing the computational cost. However, as studied in [4], the world-centric MSCKF is inconsistent. To enable consistent MSCKF-based 3D localization, a novel lightweight, robocentric VIO (R-VIO) algorithm is proposed in this paper with the following highlights:

- The global frame is treated as a feature which involves the *gravity* effect, while the local frame of reference is shifted at every image time through a *composition* step.
- The relative motion estimates used for updating global pose are obtained by *tightly* fusing the visual and inertial measurements in a local frame of reference, for which, instead of the features, a sliding *relative pose* window is included in the state to reduce the computational cost.
- An efficient *inverse-depth* measurement model is used, which allows for stably fusing the information provided by the features at infinity, even sensors stay motionless.
- A *constant* unobservable subspace is analytically shown by using the proposed robocentric formulation, which is independent of the linearization points while possessing correct dimensions *and* desired unobservable directions, significantly improving the consistency of estimation.

We perform extensive tests on both Monte Carlo simulations and real-world experiments running real data of different sensor platforms, from micro aerial vehicle (MAV) flying indoor to motor vehicle driving in dynamic traffic scenarios. All results are obtained in real time and validate the superior performance of the proposed R-VIO algorithm.

The reminder of the paper is organized as follows: We present in detail the proposed R-VIO algorithm in Section II. We demonstrate the superior performance of the proposed algorithm against the state-of-the-art VINS algorithms through both simulations and realtime experiments in Sections III and IV. In Section V, we conclude the work in this paper, as well as the possible future research directions.

## II. Robocentric Visual-Inertial Odometry

In this section, we deliberately reformulate the 3D VINS problem with respect to a moving local, rather than the fixed global, frame of reference and present in detail the proposed robocentric VIO, in comparison to the standard world-centric MSCKF [1]. In particular, we shift local frame of reference at every image time through the composition step which is not needed in the MSCKF, and analytically show that using the proposed robocentric formulation the resulting R-VIO does not have the inconsistency issue [4], [9], [10] encountered in the world-centric systems.

### A. State vector

The proposed robocentric state vector consists of: (i) the global state maintaining the information of the starting frame $\{G\}$ (i.e., $\{R_0\}$), and (ii) the IMU state characterizing the motion from the local frame of reference to the current IMU frame. In particular, at time $t_\tau \in [t_k, t_{k+1}]$, the state vector expressed in $\{R_k\}$ is given by:[1]

$$
\begin{aligned}
{}^{R_k}\mathbf{x}_\tau &= \begin{bmatrix} {}^{R_k}\mathbf{x}_G^\top & {}^{R_k}\mathbf{x}_{I_\tau}^\top \end{bmatrix}^\top \\
{}^{R_k}\mathbf{x}_G &= \begin{bmatrix} {}^{k}_{G}\bar{q}^\top & {}^{R_k}\mathbf{p}_G^\top & {}^{R_k}\mathbf{g}^\top \end{bmatrix}^\top \\
{}^{R_k}\mathbf{x}_{I_\tau} &= \begin{bmatrix} {}^{\tau}_{k}\bar{q}^\top & {}^{R_k}\mathbf{p}_{I_\tau}^\top & \mathbf{v}_{I_\tau}^\top & \mathbf{b}_{g_\tau}^\top & \mathbf{b}_{a_\tau}^\top \end{bmatrix}^\top
\end{aligned} \tag{1}
$$

where ${}^{k}_{G}\bar{q}$ is the unit quaternion [17] describing the rotation from $\{G\}$ to $\{R_k\}$, ${}^{R_k}\mathbf{p}_G$ is the position of $\{G\}$ in $\{R_k\}$, ${}^{\tau}_{k}\bar{q}$ and ${}^{R_k}\mathbf{p}_{I_\tau}$ are the relative rotation and translation from $\{R_k\}$ to $\{I_\tau\}$, and $\mathbf{v}_{I_\tau}$ is the velocity in the IMU frame. $\mathbf{b}_{g_\tau}$ and $\mathbf{b}_{a_\tau}$ denote the IMU's gyroscope and accelerometer biases, respectively. Note that the local gravity, ${}^{R_k}\mathbf{g}$, is also jointly estimated. The corresponding error state is given by:

$$
\begin{aligned}
{}^{R_k}\tilde{\mathbf{x}}_\tau &= \begin{bmatrix} {}^{R_k}\tilde{\mathbf{x}}_G^\top & {}^{R_k}\tilde{\mathbf{x}}_{I_\tau}^\top \end{bmatrix}^\top \\
{}^{R_k}\tilde{\mathbf{x}}_G &= \begin{bmatrix} \delta\boldsymbol{\theta}_G^\top & {}^{R_k}\tilde{\mathbf{p}}_G^\top & {}^{R_k}\tilde{\mathbf{g}}^\top \end{bmatrix}^\top \\
{}^{R_k}\tilde{\mathbf{x}}_{I_\tau} &= \begin{bmatrix} \delta\boldsymbol{\theta}_\tau^\top & {}^{R_k}\tilde{\mathbf{p}}_{I_\tau}^\top & \tilde{\mathbf{v}}_{I_\tau}^\top & \tilde{\mathbf{b}}_{g_\tau}^\top & \tilde{\mathbf{b}}_{a_\tau}^\top \end{bmatrix}^\top
\end{aligned} \tag{2}
$$

In particular, the error quaternion is defined by $\bar{q} = \delta\bar{q} \otimes \hat{\bar{q}}$:

$$
\delta\bar{q} \simeq \begin{bmatrix} \frac{1}{2}\delta\boldsymbol{\theta}^\top & 1 \end{bmatrix}^\top, \quad \mathbf{C}(\delta\bar{q}) = \mathbf{I}_3 - \lfloor \delta\boldsymbol{\theta} \times \rfloor \tag{3}
$$

where $\otimes$ denotes the quternion multiplication, $\delta\bar{q}$ is the error quaternion associated with 3DOF error angle $\delta\boldsymbol{\theta}$, and $\mathbf{C}(\cdot)$ denotes a $3 \times 3$ rotation matrix with $\lfloor \cdot \times \rfloor$ being the skew-symmetric operator [18].

At time-step $k$ when $\{I_k\}$ becomes the frame of reference (i.e., $\{R_k\}$), a window of the relative poses between the last $N$ robocentric frames of reference is included in the state:

$$
\begin{aligned}
\hat{\mathbf{x}}_k &= \begin{bmatrix} {}^{R_k}\hat{\mathbf{x}}_k^\top & \hat{\mathbf{w}}_k^\top \end{bmatrix}^\top \\
\hat{\mathbf{w}}_k &= \begin{bmatrix} {}^{2}_{1}\hat{\bar{q}}^\top & {}^{R_1}\hat{\mathbf{p}}_{R_2}^\top & \cdots & {}^{N}_{N-1}\hat{\bar{q}}^\top & {}^{R_{N-1}}\hat{\mathbf{p}}_{R_N}^\top \end{bmatrix}^\top
\end{aligned} \tag{4}
$$

where ${}^{i}_{i-1}\hat{\bar{q}}$ and ${}^{R_{i-1}}\hat{\mathbf{p}}_{R_i}$, $i = 2, \ldots, N$ are the estimated relative rotation and translation from $\{R_{i-1}\}$ to $\{R_i\}$. In our implementation, we manage this in a sliding-window fashion – marginalize the oldest relative pose out when the new one is cloned, to save the computational cost.

### B. Propagation

In order to use EKF, we first present the motion model for the robocentric state, ${}^{R_k}\mathbf{x}_\tau$, then extend it to the augmented state, $\mathbf{x}_\tau$. Note that the global frame is static with respect to the local frame of reference, $\{R_k\}$, i.e., ${}^{R_k}\dot{\tilde{\mathbf{x}}}_G = \mathbf{0}_{9\times 1}$.

[1] Throughout this paper, $k, k+1, \ldots$ indicate the image time-steps, while $\tau, \tau+1, \ldots$ are the IMU time-steps between every two consecutive images. $\{I\}$ and $\{C\}$ denote the IMU frame and camera frame, respectively, $\{R\}$ is the robocentric frame of reference which is selected with the corresponding IMU frame at every image time-step. The subscript $\ell|i$ refers to the estimate of a quantity at time-step $\ell$, after all measurements up to time-step $i$ have been processed. $\hat{x}$ is used to denote the estimate of a random variable $x$, while $\tilde{x} = x - \hat{x}$ is the additive error in this estimate. $\mathbf{I}_n$ and $\mathbf{0}_n$ are the $n \times n$ identity and zero matrices, respectively. Finally, the left superscript denotes the frame of reference with respect to which the vector is expressed.

For IMU state evolution, we introduce a local-parameterized kinematic model (see [19] for more details):

$$\begin{aligned} &{}^{\tau}_{k}\dot{\bar{q}} = \frac{1}{2}\mathbf{\Omega}(\boldsymbol{\omega}){}^{\tau}_{k}\bar{q},\ {}^{R_k}\dot{\mathbf{p}}_{I_\tau} = \mathbf{C}({}^{\tau}_{k}\bar{q})^{\top}\mathbf{v}_{I_\tau}, \\ &\dot{\mathbf{v}}_{I_\tau} = {}^{\tau}\mathbf{a} - \lfloor\boldsymbol{\omega}\times\rfloor\mathbf{v}_{I_\tau},\ \dot{\mathbf{b}}_g = \mathbf{n}_{wg},\ \dot{\mathbf{b}}_a = \mathbf{n}_{wa} \end{aligned} \tag{5}$$

where $\mathbf{n}_{wg}$ and $\mathbf{n}_{wa}$ denote the zero-mean white Gaussian noises that drive the IMU biases, while $\boldsymbol{\omega}$ and ${}^{\tau}\mathbf{a}$ are the angular velocity and linear acceleration expressed in $\{I_\tau\}$, respectively.

Typically, IMU provides the gyroscope and accelerometer measurements, $\boldsymbol{\omega}_m$ and $\mathbf{a}_m$, expressed in the IMU frame:

$$\boldsymbol{\omega}_m = \boldsymbol{\omega} + \mathbf{b}_g + \mathbf{n}_g \tag{6}$$

$$\mathbf{a}_m = {}^{I}\mathbf{a} + {}^{I}\mathbf{g} + \mathbf{b}_a + \mathbf{n}_a \tag{7}$$

where $\mathbf{n}_g$ and $\mathbf{n}_a$ are the zero-mean white Gaussian noises, and ${}^{I}\mathbf{g}$ characterizes the gravity effect to the sensor frame.

Linearizing (5) about the current state estimate yields the following continuous-time IMU state propagation:

$$\begin{aligned} &{}^{\tau}_{k}\dot{\hat{\bar{q}}} = \frac{1}{2}\mathbf{\Omega}(\hat{\boldsymbol{\omega}}){}^{\tau}_{k}\hat{\bar{q}},\ {}^{R_k}\dot{\hat{\mathbf{p}}}_{I_\tau} = {}^{\tau}_{k}\mathbf{C}^{\top}_{\hat{\bar{q}}}\hat{\mathbf{v}}_{I_\tau}, \\ &\dot{\hat{\mathbf{v}}}_{I_\tau} = \hat{\mathbf{a}} - {}^{\tau}\hat{\mathbf{g}} - \lfloor\hat{\boldsymbol{\omega}}\times\rfloor\hat{\mathbf{v}}_{I_\tau},\ \dot{\hat{\mathbf{b}}}_g = \mathbf{0}_{3\times1},\ \dot{\hat{\mathbf{b}}}_a = \mathbf{0}_{3\times1} \end{aligned} \tag{8}$$

where for brevity we have denoted $\hat{\boldsymbol{\omega}} = \boldsymbol{\omega}_m - \hat{\mathbf{b}}_g$, $\hat{\mathbf{a}} = \mathbf{a}_m - \hat{\mathbf{b}}_a$, and ${}^{\tau}_{k}\mathbf{C}_{\hat{\bar{q}}} = \mathbf{C}({}^{\tau}_{k}\hat{\bar{q}})$. Thus, the continuous-time linearized robocentric error-state model can be obtained as:

$$\begin{bmatrix} {}^{R_k}\dot{\tilde{\mathbf{x}}}_G \\ {}^{R_k}\dot{\tilde{\mathbf{x}}}_{I_\tau} \end{bmatrix} = \begin{bmatrix} \mathbf{0}_{9\times24} \\ \mathbf{F}_I \end{bmatrix} \begin{bmatrix} {}^{R_k}\tilde{\mathbf{x}}_G \\ {}^{R_k}\tilde{\mathbf{x}}_{I_\tau} \end{bmatrix} + \begin{bmatrix} \mathbf{0}_{9\times12} \\ \mathbf{G}_I \end{bmatrix} \mathbf{n} \tag{9}$$

$$\Rightarrow\ {}^{R_k}\dot{\tilde{\mathbf{x}}}_\tau = \mathbf{F}{}^{R_k}\tilde{\mathbf{x}}_\tau + \mathbf{G}\mathbf{n} \tag{10}$$

where $\mathbf{n} = [\mathbf{n}_g^{\top} \quad \mathbf{n}_{wg}^{\top} \quad \mathbf{n}_a^{\top} \quad \mathbf{n}_{wa}^{\top}]^{\top}$ is the input noise vector, and $\mathbf{F}_I$ and $\mathbf{G}_I$ are the continuous-time IMU error-state transition matrix and noise Jacobian (see [19]), respectively.

As for an actual implementation of EKF, the discrete-time propagation model is needed. First, the IMU state estimate, ${}^{R_k}\hat{\mathbf{x}}_{I_{\tau+1}}$, is obtained as follows: (i) compute ${}^{\tau+1}_{k}\hat{\bar{q}}$ using the zeroth order quaternion integrator [18]; (ii) calculate ${}^{R_k}\hat{\mathbf{p}}_{I_{\tau+1}}$ and ${}^{R_k}\hat{\mathbf{v}}_{I_{\tau+1}}$ using the IMU preintegration:

$${}^{R_k}\hat{\mathbf{p}}_{I_{\tau+1}} = {}^{R_k}\hat{\mathbf{p}}_{I_\tau} + \Delta\mathbf{p}_{\tau,\tau+1} \tag{11}$$

$${}^{R_k}\hat{\mathbf{v}}_{I_{\tau+1}} = {}^{R_k}\hat{\mathbf{v}}_{I_\tau} + \Delta\mathbf{v}_{\tau,\tau+1} \tag{12}$$

where the preintegrated terms, $\Delta\mathbf{p}$ and $\Delta\mathbf{v}$, can be recursively computed with the incoming IMU measurements [20]. Therefore, the local velocity estimate, $\hat{\mathbf{v}}_{I_{\tau+1}}$, can be obtained using the result of (i): $\hat{\mathbf{v}}_{I_{\tau+1}} = {}^{\tau+1}_{k}\mathbf{C}_{\hat{\bar{q}}}{}^{R_k}\hat{\mathbf{v}}_{I_{\tau+1}}$; (iii) assume the bias estimates are constant in the time interval $[t_\tau, t_{\tau+1}]$: $\hat{\mathbf{b}}_{g_{\tau+1}} = \hat{\mathbf{b}}_{g_\tau}$, and $\hat{\mathbf{b}}_{a_{\tau+1}} = \hat{\mathbf{b}}_{a_\tau}$.

Then, for covariance propagation the discrete-time error-state transition matrix $\mathbf{\Phi}(t_{\tau+1}, t_\tau)$ can be obtained using the forward Euler method over the time interval $[t_\tau, t_{\tau+1}]$:

$$\mathbf{\Phi}(t_{\tau+1}, t_\tau) = \exp(\mathbf{F}\delta t) \simeq \mathbf{I}_{24} + \mathbf{F}\delta t =: \mathbf{\Phi}_{\tau+1,\tau} \tag{13}$$

where $\delta t = t_{\tau+1} - t_\tau$. It results in the covariance propagation starting from time-step $k$ (see [19] for more details):

$$\mathbf{P}_{\tau+1|k} = \mathbf{\Phi}_{\tau+1,\tau}\mathbf{P}_{\tau|k}\mathbf{\Phi}^{\top}_{\tau+1,\tau} + \mathbf{G}\mathbf{Q}\mathbf{G}^{\top} \tag{14}$$

where $\mathbf{Q} = \mathbf{Diag}\begin{bmatrix}\delta t\sigma_g^2\mathbf{I}_3 & \delta t\sigma_{wg}^2\mathbf{I}_3 & \delta t\sigma_a^2\mathbf{I}_3 & \delta t\sigma_{wa}^2\mathbf{I}_3\end{bmatrix}$ denotes the discrete-time input noise covariance matrix. If we partition the augmented covariance matrix at time-step $k$ according to the robocentric and sliding-window states in (4): $\mathbf{P}_k = \begin{bmatrix}\mathbf{P}_{\mathbf{x}\mathbf{x}_k} & \mathbf{P}_{\mathbf{x}\mathbf{w}_k} \\ \mathbf{P}^{\top}_{\mathbf{x}\mathbf{w}_k} & \mathbf{P}_{\mathbf{w}\mathbf{w}_k}\end{bmatrix}$. By noting that the relative poses in sliding window are of zero kinematics: $\hat{\mathbf{w}}_{\tau+1} = \hat{\mathbf{w}}_\tau \equiv \hat{\mathbf{w}}_k$, the propagated covariance matrix at time-step $\tau+1$ is given by:

$$\mathbf{P}_{\tau+1|k} = \begin{bmatrix} \mathbf{P}_{\mathbf{x}\mathbf{x}_{\tau+1|k}} & \mathbf{\Phi}_{\tau+1,k}\mathbf{P}_{\mathbf{x}\mathbf{w}_k} \\ \mathbf{P}^{\top}_{\mathbf{x}\mathbf{w}_k}\mathbf{\Phi}^{\top}_{\tau+1,k} & \mathbf{P}_{\mathbf{w}\mathbf{w}_k} \end{bmatrix} \tag{15}$$

where $\mathbf{P}_{\mathbf{x}\mathbf{x}_{\tau+1|k}}$ can be recursively computed using (14), and the compound error-state transition matrix is computed as: $\mathbf{\Phi}_{\tau+1,k} = \prod_{\ell=k}^{\tau}\mathbf{\Phi}_{\ell+\delta t,\ell}$, with $\mathbf{\Phi}_{k,k} = \mathbf{I}_{24}$ at time-step $k$.

### C. Update

*1) Inverse-depth measurement model:* In order for robust update, we adopt the *inverse depth* parameterization [21] for landmarks observed from a monocular camera, while being *tailored* for the proposed robocentric VIO. Assuming a single landmark $L_j$ has been observed from a set of $n_j$ robocentric frames, $\mathcal{R}_j$, the measurement of $L_j$ is given by the following bearing-only model in the $xyz$ coordinates ($i \in \mathcal{R}_j$):

$$\mathbf{z}_{j,i} = \frac{1}{z^i_j}\begin{bmatrix} x^i_j \\ y^i_j \end{bmatrix} + \mathbf{n}_{j,i}, \quad {}^{C_i}\mathbf{p}_{L_j} = \begin{bmatrix} x^i_j \\ y^i_j \\ z^i_j \end{bmatrix} \tag{16}$$

where $\mathbf{n}_{j,i} \sim \mathcal{N}(\mathbf{0}, \sigma^2_{im}\mathbf{I}_2)$ represents the image noise, and ${}^{C_i}\mathbf{p}_{L_j}$ is the position of $L_j$ in $\{C_i\}$, associated with $\{R_i\}$. With that, the inverse-depth form for ${}^{C_i}\mathbf{p}_{L_j}$ is given by:

$${}^{C_i}\mathbf{p}_{L_j} = {}^{i}_{1}\bar{\mathbf{C}}_{\bar{q}}{}^{C_1}\mathbf{p}_{L_j} + {}^{i}\bar{\mathbf{p}}_1 =: \mathbf{f}_i(\phi, \psi, \rho) \tag{17}$$

$${}^{C_1}\mathbf{p}_{L_j} = \frac{1}{\rho}\mathbf{e}(\phi, \psi), \quad \mathbf{e} = \begin{bmatrix} \cos\phi\sin\psi \\ \sin\phi \\ \cos\phi\cos\psi \end{bmatrix} \tag{18}$$

where ${}^{C_1}\mathbf{p}_{L_j}$ is the position of $L_j$ in the first camera frame of $\mathcal{R}_j$, $\mathbf{e}$ is the directional vector with $\phi$ and $\psi$ the elevation and azimuth expressed in $\{C_1\}$, and $\rho$ is the inverse depth along $\mathbf{e}$. In particular, the relative poses between $\{C_1\}$ and $\{C_i\}$, $i \in \mathcal{R}_j \backslash 1$ are expressed using the camera-to-IMU calibration parameters, $\{{}^{C}_{I}\bar{q}, {}^{C}\mathbf{p}_I\}$, and the sliding-window state, $\mathbf{w}$, as:

$${}^{i}_{1}\bar{\mathbf{C}}_{\bar{q}} = {}^{C}_{I}\mathbf{C}_{\bar{q}1}{}^{i}_{1}\mathbf{C}_{\bar{q}C}{}^{I}_{C}\mathbf{C}_{\bar{q}} \tag{19}$$

$${}^{i}\bar{\mathbf{p}}_1 = {}^{C}_{I}\mathbf{C}_{\bar{q}1}{}^{i}_{1}\mathbf{C}_{\bar{q}}{}^{I}\mathbf{p}_C + {}^{C}_{I}\mathbf{C}_{\bar{q}}{}^{R_i}\mathbf{p}_{R_1} + {}^{C}\mathbf{p}_I \tag{20}$$

Interestingly, if the landmark is at infinity (i.e., $\rho \to 0$), we can normalize (17) by premultiplying $\rho$ to avoid numerical issues, i.e.,

$$\begin{aligned} \rho{}^{C_i}\mathbf{p}_{L_j} &= {}^{i}_{1}\bar{\mathbf{C}}_{\bar{q}}\mathbf{e}(\phi, \psi) + \rho{}^{i}\bar{\mathbf{p}}_1 \\ &=: \mathbf{h}_i(\mathbf{w}, \phi, \psi, \rho) = \begin{bmatrix} h_{i,1}(\mathbf{w}, \phi, \psi, \rho) \\ h_{i,2}(\mathbf{w}, \phi, \psi, \rho) \\ h_{i,3}(\mathbf{w}, \phi, \psi, \rho) \end{bmatrix} \end{aligned} \tag{21}$$

Note that, this equation reserves the geometry of (17) and encompasses two degenerate cases: (i) observing the landmarks

at infinity (i.e., $\rho \to 0$), and (ii) having low parallax between two camera poses (i.e., ${}^{i}\bar{\mathbf{p}}_1 \to 0$). For both cases, (21) can be approximated by $\mathbf{h}_i \simeq {}^{i}_{1}\bar{\mathbf{C}}_{\bar{q}}\mathbf{e}(\phi,\psi)$, and hence the corresponding measurements can still provide the information about the orientation. Therefore, we introduce the following inverse-depth measurement model for the robocentric VIO:

$$\mathbf{z}_{j,i} = \frac{1}{h_{i,3}(\mathbf{w},\phi,\psi,\rho)}\begin{bmatrix} h_{i,1}(\mathbf{w},\phi,\psi,\rho) \\ h_{i,2}(\mathbf{w},\phi,\psi,\rho)\end{bmatrix} + \mathbf{n}_{j,i} \tag{22}$$

Denoting $\boldsymbol{\lambda} = [\phi,\psi,\rho]^\top$ and linearizing (22) at current state estimates, $\hat{\mathbf{x}}$, and $\hat{\boldsymbol{\lambda}}$, we have:

$$\mathbf{r}_{j,i} = \mathbf{z}_{j,i} - \hat{\mathbf{z}}_{j,i} \simeq \mathbf{H}_{\mathbf{x}_{j,i}}\tilde{\mathbf{x}} + \mathbf{H}_{\boldsymbol{\lambda}_{j,i}}\tilde{\boldsymbol{\lambda}} + \mathbf{n}_{j,i} \tag{23}$$

$$\mathbf{H}_{\mathbf{x}_{j,i}} = \mathbf{H}_{\mathrm{p}_{j,i}}\begin{bmatrix}\mathbf{0}_{3\times 24} & \mathbf{H}_{\mathbf{w}_{j,i}}\end{bmatrix},$$

$$\mathbf{H}_{\boldsymbol{\lambda}_{j,i}} = \mathbf{H}_{\mathrm{p}_{j,i}}\mathbf{H}_{\mathrm{inv}_{j,i}}, \quad \mathbf{H}_{\mathrm{p}_{j,i}} = \frac{1}{\hat{h}_{i,3}}\begin{bmatrix}1 & 0 & -\frac{\hat{h}_{i,1}}{\hat{h}_{i,3}} \\ 0 & 1 & -\frac{\hat{h}_{i,2}}{\hat{h}_{i,3}}\end{bmatrix},$$

$$\mathbf{H}_{\mathrm{inv}_{j,i}} = \frac{\partial \mathbf{h}_i}{\partial \tilde{\boldsymbol{\lambda}}} = \begin{bmatrix}\frac{\partial \mathbf{h}_i}{\partial[\tilde{\phi},\tilde{\psi}]^\top} & \frac{\partial \mathbf{h}_i}{\partial \tilde{\rho}}\end{bmatrix}, \quad \text{and}$$

$$\mathbf{H}_{\mathbf{w}_{j,i}} = \frac{\partial \mathbf{h}_i}{\partial \tilde{\mathbf{w}}} = \begin{bmatrix}\frac{\partial \mathbf{h}_i}{\partial \delta\boldsymbol{\theta}_2} & \frac{\partial \mathbf{h}_i}{\partial {}^{R_1}\tilde{\mathbf{p}}_{R_2}} & \cdots & \frac{\partial \mathbf{h}_i}{\partial \delta\boldsymbol{\theta}_N} & \frac{\partial \mathbf{h}_i}{\partial {}^{R_{N-1}}\tilde{\mathbf{p}}_{R_N}}\end{bmatrix},$$

$$\frac{\partial \mathbf{h}_i}{\partial \delta\boldsymbol{\theta}_n} = {}^{C}_{I}\mathbf{C}_{\bar{q}}\,{}^{i}_{1}\mathbf{C}_{\hat{\bar{q}}}\lfloor\big({}^{I}_{C}\mathbf{C}_{\bar{q}}\hat{\mathbf{e}} + \hat{\rho}\,{}^{I}\mathbf{p}_C - \hat{\rho}\,{}^{R_1}\hat{\mathbf{p}}_{R_n}\big)\times\rfloor\,{}^{n}_{1}\mathbf{C}^\top_{\hat{\bar{q}}},$$

$$\frac{\partial \mathbf{h}_i}{\partial {}^{R_{n-1}}\tilde{\mathbf{p}}_{R_n}} = -\hat{\rho}\,{}^{C}_{I}\mathbf{C}_{\bar{q}}\,{}^{i}_{n-1}\mathbf{C}_{\hat{\bar{q}}}, \quad n = 2,\ldots,i \le N \tag{24}$$

where $\mathbf{H}_{\mathbf{x}_{j,i}}$ and $\mathbf{H}_{\boldsymbol{\lambda}_{j,i}}$ are the Jacobians with respect to the vectors of states and inverse-depth parameters, respectively. Since an estimate of $\boldsymbol{\lambda}$ is needed for computing (23), a particular bundle adjustment is solved first using the measurements $\mathbf{z}_{j,i}$, $i \in \mathcal{R}_j$, and the relative pose estimates, $\hat{\mathbf{w}}$ (see [19]). By stacking the residual $\mathbf{r}_{j,i}$, $i \in \mathcal{R}_j$, we obtain:

$$\mathbf{r}_j \simeq \mathbf{H}_{\mathbf{x}_j}\tilde{\mathbf{x}} + \mathbf{H}_{\boldsymbol{\lambda}_j}\tilde{\boldsymbol{\lambda}} + \mathbf{n}_j \tag{25}$$

assuming the measurements obtained from different camera poses are independent, the covariance matrix of $\mathbf{n}_j$ is hence $\mathbf{R}_j = \sigma^2_{im}\mathbf{I}_{2n_j}$. As $\hat{\mathbf{x}}$ (precisely, $\hat{\mathbf{w}}$) is used to compute $\hat{\boldsymbol{\lambda}}$, the inverse-depth error $\tilde{\boldsymbol{\lambda}}$ is correlated to $\tilde{\mathbf{x}}$. In order to find a valid residual for EKF update, we project (25) to the left nullspace of $\mathbf{H}_{\boldsymbol{\lambda}_j}$ (i.e., $\mathbf{O}^\top_{\boldsymbol{\lambda}_j}\mathbf{H}_{\boldsymbol{\lambda}_j} = \mathbf{0}$, and $\mathbf{O}^\top_{\boldsymbol{\lambda}_j}\mathbf{O}_{\boldsymbol{\lambda}_j} = \mathbf{I}$):

$$\bar{\mathbf{r}}_j = \mathbf{O}^\top_{\boldsymbol{\lambda}_j}\mathbf{r}_j \simeq \mathbf{O}^\top_{\boldsymbol{\lambda}_j}\mathbf{H}_{\mathbf{x}_j}\tilde{\mathbf{x}} + \mathbf{O}^\top_{\boldsymbol{\lambda}_j}\mathbf{n}_j = \bar{\mathbf{H}}_{\mathbf{x}_j}\tilde{\mathbf{x}} + \bar{\mathbf{n}}_j \tag{26}$$

for which we discussed the possible rank deficiency in $\mathbf{H}_{\boldsymbol{\lambda}_j}$ (see [19] for more details), while in general case this $2n_j \times 3$ matrix has full rank with the nullspace of dimension $2n_j - 3$. And for (26), we use *Givens rotations* [22], with complexity $O(n_j^2)$. Since $\mathbf{O}_{\boldsymbol{\lambda}_j}$ is unitary, the covariance matrix of $\bar{\mathbf{n}}_j$ is:

$$\bar{\mathbf{R}}_j = \mathbf{O}^\top_{\boldsymbol{\lambda}_j}\mathbf{R}_j\mathbf{O}_{\boldsymbol{\lambda}_j} = \sigma^2_{im}\mathbf{I}_{2n_j-3} \tag{27}$$

Note that, in (24) each measurement of $L_j$ is correlated to a chain of the relative poses in $\mathbf{w}$, which forms a *dense* connection between the measurements and the state. This is, compared to [1] where each measurement is only correlated to the pose from which it is observed, more efficient without increasing the computational complexity. In addition, prior to an EKF update, the Mahalanobis distance for each landmark is checked, which serves as the probabilistic outlier rejection.

*2) EKF update:* Assuming that at time-step $k+1$ we have the measurements of $M$ landmarks to process, we can stack the resulting $\bar{\mathbf{r}}_j$, $j = 1,\ldots,M$, to have:

$$\bar{\mathbf{r}} = \bar{\mathbf{H}}_{\mathbf{x}}\tilde{\mathbf{x}} + \bar{\mathbf{n}} \tag{28}$$

of which the dimension is $d = \sum_{j=1}^{M}(2n_j - 3)$. However, in practice $d$ tends to be a large number even though $M$ is small (e.g., $d = 170$, if 10 landmarks are observed from 10 robot poses). To reduce the computational complexity, QR decomposition is applied to this measurement model before doing an EKF update. We note that $\bar{\mathbf{H}}_{\mathbf{x}}$ is *rank deficient* with zero columns corresponding to the robocentric state, while the nonzero columns corresponding to the relative poses are linearly independent. Therefore, the QR decomposition can be applied to the nonzero part of $\bar{\mathbf{H}}_{\mathbf{x}}$ only, as:

$$\begin{aligned}\bar{\mathbf{H}}_{\mathbf{x}} &= \begin{bmatrix}\mathbf{0}_{d\times 24} & \bar{\mathbf{H}}_{\mathbf{w}}\end{bmatrix} \\ &= \begin{bmatrix}\mathbf{0}_{d\times 24} & \begin{bmatrix}\mathbf{Q}_1 & \mathbf{Q}_2\end{bmatrix}\begin{bmatrix}\bar{\mathbf{T}}_{\mathbf{w}} \\ \mathbf{0}_{(d-6(N-1))\times 6(N-1)}\end{bmatrix}\end{bmatrix} \\ &= \begin{bmatrix}\mathbf{Q}_1 & \mathbf{Q}_2\end{bmatrix}\begin{bmatrix}\mathbf{0}_{d\times 24} & \begin{bmatrix}\bar{\mathbf{T}}_{\mathbf{w}} \\ \mathbf{0}_{(d-6(N-1))\times 6(N-1)}\end{bmatrix}\end{bmatrix}\end{aligned} \tag{29}$$

where $\mathbf{Q}_1$ and $\mathbf{Q}_2$ are two unitary matrices, and $\bar{\mathbf{T}}_{\mathbf{w}}$ is an upper triangular matrix. With this definition, (28) yields:

$$\bar{\mathbf{r}} = \begin{bmatrix}\mathbf{Q}_1 & \mathbf{Q}_2\end{bmatrix}\begin{bmatrix}\mathbf{0} & \bar{\mathbf{T}}_{\mathbf{w}} \\ \mathbf{0} & \mathbf{0}\end{bmatrix}\tilde{\mathbf{x}} + \bar{\mathbf{n}} \tag{30}$$

$$\Rightarrow \begin{bmatrix}\mathbf{Q}_1^\top \\ \mathbf{Q}_2^\top\end{bmatrix}\bar{\mathbf{r}} = \begin{bmatrix}\mathbf{0} & \bar{\mathbf{T}}_{\mathbf{w}} \\ \mathbf{0} & \mathbf{0}\end{bmatrix}\tilde{\mathbf{x}} + \begin{bmatrix}\mathbf{Q}_1^\top \\ \mathbf{Q}_2^\top\end{bmatrix}\bar{\mathbf{n}} \tag{31}$$

for which we discard the lower rows which are only about the measurement noise, and employ the upper $6(N-1)$ rows, instead of (28), to be the residual for EKF update:

$$\breve{\mathbf{r}} = \mathbf{Q}_1^\top\bar{\mathbf{r}} = \begin{bmatrix}\mathbf{0} & \bar{\mathbf{T}}_{\mathbf{w}}\end{bmatrix}\tilde{\mathbf{x}} + \mathbf{Q}_1^\top\bar{\mathbf{n}} = \breve{\mathbf{H}}_{\mathbf{x}}\tilde{\mathbf{x}} + \breve{\mathbf{n}} \tag{32}$$

where $\breve{\mathbf{n}} = \mathbf{Q}_1^\top\bar{\mathbf{n}}$ is the noise vector with the covariance matrix $\breve{\mathbf{R}} = \mathbf{Q}_1^\top\bar{\mathbf{R}}\mathbf{Q}_1 = \sigma^2_{im}\mathbf{I}_{6(N-1)}$. Especially, when we have $d \gg 6(N-1)$ these can be done by Givens rotations, with complexity $O(N^2 d)$. Next, the EKF update follows:

$$\begin{aligned}\mathbf{K} &= \mathbf{P}\breve{\mathbf{H}}_{\mathbf{x}}^\top\big(\breve{\mathbf{H}}_{\mathbf{x}}\mathbf{P}\breve{\mathbf{H}}_{\mathbf{x}}^\top + \breve{\mathbf{R}}\big)^{-1} \\ \hat{\mathbf{x}}_{k+1|k+1} &= \hat{\mathbf{x}}_{k+1|k} + \mathbf{K}\breve{\mathbf{r}} \\ \mathbf{P}_{k+1|k+1} &= \big(\mathbf{I} - \mathbf{K}\breve{\mathbf{H}}_{\mathbf{x}}\big)\mathbf{P}_{k+1|k}\big(\mathbf{I} - \mathbf{K}\breve{\mathbf{H}}_{\mathbf{x}}\big)^\top + \mathbf{K}\breve{\mathbf{R}}\mathbf{K}^\top\end{aligned}$$

*3) State augmentation:* In contrast to [1], here we perform state augmentation right after an EKF update, where a copy of the relative pose estimate, $\{{}^{k+1}_{k}\hat{\bar{q}}_{k+1|k+1}, {}^{R_k}\hat{\mathbf{p}}_{I_{k+1|k+1}}\}$, is appended to the end of the state vector, $\hat{\mathbf{x}}_{k+1|k+1}$. Accordingly, the covariance matrix is augmented following:

$$\mathbf{P}_{k+1|k+1} \leftarrow \begin{bmatrix}\mathbf{I}_{24+6(N-1)} \\ \mathbf{J}\end{bmatrix}\mathbf{P}_{k+1|k+1}\begin{bmatrix}\mathbf{I}_{24+6(N-1)} \\ \mathbf{J}\end{bmatrix}^\top,$$

$$\mathbf{J} = \begin{bmatrix}\mathbf{0}_{3\times 9} & \mathbf{I}_3 & \mathbf{0}_3 & \mathbf{0}_{3\times 9} & \mathbf{0}_{3\times 6(N-1)} \\ \mathbf{0}_{3\times 9} & \mathbf{0}_3 & \mathbf{I}_3 & \mathbf{0}_{3\times 9} & \mathbf{0}_{3\times 6(N-1)}\end{bmatrix} \tag{33}$$

In this way, we have optimal relative pose estimates included in the state, thus yielding an *optimal* state augmentation.

### D. Composition

We shift local frame of reference at every image time. At this point, the corresponding IMU frame, $\{I_{k+1}\}$, is chosen as the new frame of reference, i.e., $\{R_{k+1}\}$. The state vector expressed in $\{R_{k+1}\}$ is then obtained as:

$$\hat{\mathbf{x}}_{k+1} = \begin{bmatrix} {}^{R_{k+1}}\hat{\mathbf{x}}_{k+1} \\ \hat{\mathbf{w}}_{k+1} \end{bmatrix} = \begin{bmatrix} {}^{R_k}\hat{\mathbf{x}}_{k+1|k+1} \boxplus {}^{R_k}\hat{\mathbf{x}}_{I_{k+1|k+1}} \\ \hat{\mathbf{w}}_{k+1|k+1} \end{bmatrix}$$

$$\Rightarrow \begin{bmatrix} {}^{k+1}_{G}\hat{\bar{q}} \\ {}^{R_{k+1}}\hat{\mathbf{p}}_G \\ {}^{R_{k+1}}\hat{\mathbf{g}} \\ {}^{k+1}_{k+1}\hat{\bar{q}} \\ {}^{R_{k+1}}\hat{\mathbf{p}}_{R_{k+1}} \\ \hat{\mathbf{v}}_{R_{k+1}} \\ \hat{\mathbf{b}}_{g_{k+1}} \\ \hat{\mathbf{b}}_{a_{k+1}} \\ \hat{\mathbf{w}}_{k+1} \end{bmatrix} = \begin{bmatrix} {}^{k+1}_{k}\hat{\bar{q}} \otimes {}^{k}_{G}\hat{\bar{q}} \\ {}^{k+1}_{k}\mathbf{C}_{\hat{\bar{q}}}\big({}^{R_k}\hat{\mathbf{p}}_{G_{k+1}} - {}^{R_k}\hat{\mathbf{p}}_{I_{k+1}}\big) \\ {}^{k+1}_{k}\mathbf{C}_{\hat{\bar{q}}}{}^{R_k}\hat{\mathbf{g}} \\ \bar{q}_0 \\ \mathbf{0}_{3\times 1} \\ \hat{\mathbf{v}}_{I_{k+1}} \\ \hat{\mathbf{b}}_{g_{k+1}} \\ \hat{\mathbf{b}}_{a_{k+1}} \\ \hat{\mathbf{w}}_{k+1|k+1} \end{bmatrix} \tag{34}$$

where $\bar{q}_0 = [0, 0, 0, 1]^\top$, and for brevity we have omitted the complete subscripts in the robocentric state. Accordingly, the covariance matrix is transformed through (see [19]):

$$\mathbf{P}_{k+1} = \mathbf{U}_{k+1}\mathbf{P}_{k+1|k+1}\mathbf{U}_{k+1}^\top \tag{35}$$

$$\begin{aligned} \mathbf{U}_{k+1} &= \frac{\partial \tilde{\mathbf{x}}_{k+1}}{\partial \tilde{\mathbf{x}}_{k+1|k+1}} = \begin{bmatrix} \mathbf{V}_{k+1} & \mathbf{0}_{24\times 6N} \\ \mathbf{0}_{6N\times 24} & \mathbf{I}_{6N} \end{bmatrix}, \\ \mathbf{V}_{k+1} &= \frac{\partial {}^{R_{k+1}}\tilde{\mathbf{x}}_{k+1}}{\partial {}^{R_k}\tilde{\mathbf{x}}_{k+1|k+1}} = \begin{bmatrix} \frac{\partial {}^{R_{k+1}}\tilde{\mathbf{x}}_{k+1}}{\partial {}^{R_k}\tilde{\mathbf{x}}_{G_{k+1}}} & \frac{\partial {}^{R_{k+1}}\tilde{\mathbf{x}}_{k+1}}{\partial {}^{R_k}\tilde{\mathbf{x}}_{I_{k+1}}} \end{bmatrix} \end{aligned} \tag{36}$$

Note that, the relative pose in IMU state is *reset* to the origin, while the velocity and biases are in the sensor frame and not affected by the change of frame of reference. Specifically, in (35) the covariance of the relative pose is also *reset* to zero, i.e., no uncertainty for robocentric frame of reference itself. We outline the proposed robocentric VIO in Algorithm 1.

### E. Observability analysis

System observability reveals whether the information provided by the measurements is sufficient to estimate the state without ambiguity. To this end, we perform the observability analysis within the EKF-SLAM framework that has the same observability properties as the EKF-VIO if provided the same linearization points used [4]. For brevity of presentation, we employ the case that only a single landmark is included in the state vector, while the conclusion can be easily generalized to the case of multiple landmarks. In this case, the state vector at time $t_\ell \in [t_k, t_{k+m}]$ includes a landmark $L$:

$$\mathbf{x}_\ell = \begin{bmatrix} {}^{R_k}\mathbf{x}_G^\top & {}^{R_k}\mathbf{p}_L^\top & {}^{R_k}\mathbf{x}_{I_\ell}^\top \end{bmatrix}^\top \tag{37}$$

The measurement model (16) (or the inverse-depth model (22)) is used. The observability matrix is computed as [23]:

$$\mathbf{M} = \begin{bmatrix} \mathbf{H}_k \\ \vdots \\ \mathbf{H}_\ell \mathbf{\Psi}_{\ell,k} \\ \vdots \\ \mathbf{H}_{k+m}\mathbf{\Psi}_{k+m,k} \end{bmatrix} \tag{38}$$

**Algorithm 1** Robocentric Visual-Inertial Odometry

**Input**: camera images, and IMU measurements
**Output**: 6DOF realtime robot poses
**R-VIO**: Initialize the first frame of reference $\{R_0\}$ (=$\{G\}$) when the first IMU measurement(s) comes in. Then, when a new camera image comes in, do

- **Feature tracking**: extract visual features from the image, then perform KLT tracking and outlier rejection.
- **Propagation**: propagate states and covariance matrix using the preintegration with all IMU measurements between last and current images.
- **Update**:
  (i) *EKF update*: compute the inverse-depth measurement model matrices for the features whose track are complete (i.e., lost track, or reach the maximum tracking length). Use the features that passed Mahalanobis test for an EKF update.
  (ii) *State augmentation*: augment state and covariance matrix with the updated relative pose estimates.
- **Composition**: shift the frame of reference to current IMU frame, update global state and covariance using the relative pose estimates, and then reset relative pose (including both state and covariance).

where $\mathbf{\Psi}_{\ell,k}$ is the state transition matrix from time-step $k$ to $\ell$, and $\mathbf{H}_\ell$ is the measurement Jacobian for the observation at time-step $\ell$. It should be noted that since the composition step changes the local frame of reference, we investigate the observability for a complete cycle of the proposed robocentric EKF: propagation, update, and composition. First, we analytically compute $\mathbf{M}$ for the propagation and update, and show that the unobservable subspace (the nullspace of $\mathbf{M}$) spans the following *nine* directions in the state space:

$$\mathbf{null}(\mathbf{M}) = \underset{\text{col.}}{\text{span}} \begin{bmatrix} \mathbf{I}_3 & \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{I}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{I}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{I}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 \end{bmatrix} \tag{39}$$

Subsequently, we show that the composition dose not change the above unobservable subspace (see [19] for more details).

**Remark 1.** *(Unobservable Subspace) The first 6DOF belong to the orientation (3) and position (3) of the global frame, while the last 3DOF correspond to the same translation (3) simultaneously applied to the landmark(s) and sensor.*

**Remark 2.** *(Consistency) The unobservable subspace is constant, i.e., independent of the linearization points, which not only guarantees the correct unobservable dimensions, but the desired unobservable directions, thus improving consistency.*

### F. Initialization

It is important to point out that in the proposed robocentric formulation, the filter initialization becomes very easy, as the states are simply relative to a local frame of reference and typically start from zero *without* the need to align the initial pose with a fixed global frame. In particular, in our implementation, (i) the initial global pose and IMU relative pose are both set to $\{\bar{q}_0, \mathbf{0}_{3\times1}\}$, (ii) the initial local gravity is the average of first available accelerometer measurement(s) before moving, and (iii) the initial acceleration bias is obtained by removing gravity effects while the initial gyroscope bias is the average of the corresponding stationary measurements. Similarly, the corresponding uncertainties for the poses are set to zero, while for the local gravity and biases are set to: $\mathbf{\Sigma}_g = \Delta T \sigma_a^2 \mathbf{I}_3$, $\mathbf{\Sigma}_{b_g} = \Delta T \sigma_{wg}^2 \mathbf{I}_3$, and $\mathbf{\Sigma}_{b_a} = \Delta T \sigma_{wa}^2 \mathbf{I}_3$, where $\Delta T$ is the time length for initialization.

## III. Simulation Results

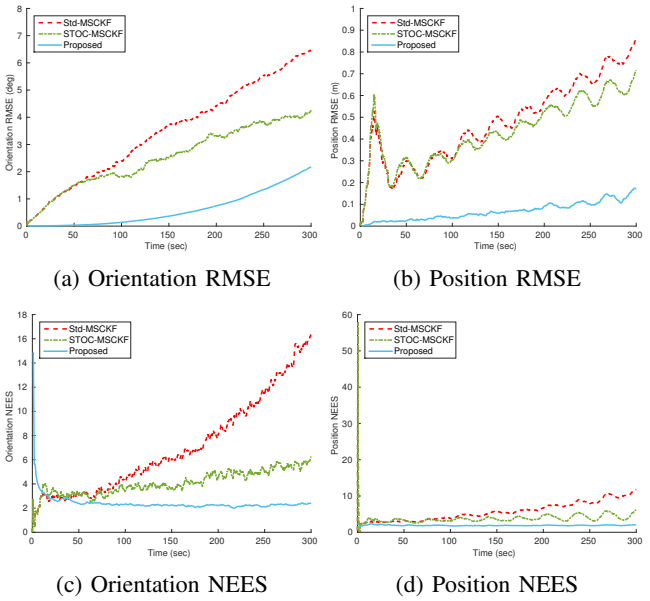
We conducted a series of Monte Carlo simulations under realistic conditions to validate the proposed algorithm. Two metrics are used for evaluation: (i) the root mean squared error (RMSE) and (ii) the normalized estimation error squared (NEES). The RMSE provides a concise metric of the accuracy of a given filter, while the NEES is a standard criterion for evaluating the filter's consistency [24]. We compared with two world-centric counterparts: the standard MSCKF [1], and the state-of-the-art state-transition observability constrained (STOC)-MSCKF [11] that enforces correct dimension of the unobservable subspace to improve consistency. To ensure a fair comparison, we implemented all filters using the same parameters, such as the sliding-window size, and processed the same data (i.e., 50 trails at MEMS sensor quality [11]). The comparison results are shown in Figure 1, and Table I provides the average RMSE and NEES for all the algorithms, which clearly show that the proposed R-VIO significantly outperforms the standard MSCKF as well as STOC-MSCKF in terms of both RMSE and NEES, attributed to the novel reformulation and consistency of the system. Note that, in Figure 1c the orientation NEES of R-VIO has a jump at the beginning which is primarily due to the small covariance we used for initialization (see Section II-F), while it can quickly recover and perform consistently only after a short time.

TABLE I: Avg. RMSE and NEES corresponding to Fig. 1

| | Orientation RMSE (°) | Position RMSE (m) | Orientation NEES | Position NEES |
|---|---|---|---|---|
| Std-MSCKF | 3.4700 | 0.4774 | 7.0487 | 5.8103 |
| STOC-MSCKF | 2.5232 | 0.4305 | 4.0964 | 3.7936 |
| R-VIO | **0.6811** | **0.0715** | **2.4146** | **1.9061** |

## IV. Experimental Results

We further experimentally validate the proposed R-VIO in both indoor and outdoor environments, using both the public benchmark dataset on micro aerial vehicle (MAV) and the urban driving dataset collected with our own sensor platform on a car. As described in Algorithm 1, we implemented a C++ multithread framework. In *front end*, the visual tracking thread extracts features from the image using the Shi-Tomasi corner detector [25], and then tracks them between pairwise images using the KLT algorithm [26]. To remove the outliers from tracking result, we integrated an IMU-aided two-point RANSAC algorithm [27]. Once the visual tracking is done, the *back end* processes all visual and inertial measurements using the proposed robocentric EKF. Especially, for the feature lost track we use all its measurements within the sliding window for an EKF update, while for the one reaching the maximum tracking length (e.g., the sliding-window size) we use its subset (e.g., 1/2) of measurements and maintain the rest for next update. All the tests run on a Core i7-4710MQ @ 2.5GHz laptop at *real time*.




(a) Orientation RMSE (b) Position RMSE

(c) Orientation NEES (d) Position NEES

Fig. 1: Statistical results of 50 Monte Carlo simulations.

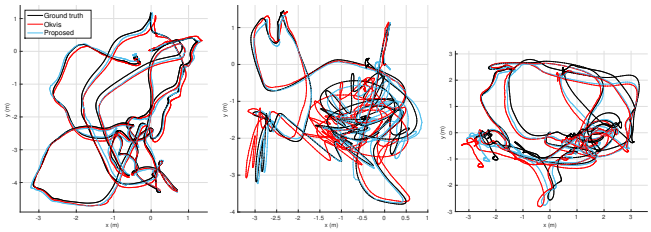
### A. EuRoC dataset

We tested R-VIO over all 11 sequences in the EuRoC dataset [28], where a FireFly helicopter equipped with VI-sensor (IMU @ 200Hz, and cameras @ 20Hz) was used for data collection. During the tests, only the left camera images were used for vision inputs and 200 features were uniformly extracted from each image. The sliding window size was set up to 20 (i.e., about 1s relative motion memory). We compared R-VIO against the OKVIS[2], a state-of-the-art world-centric keyframe-based visual-inertial SLAM system [5]. The RMSE results after 6DOF pose alignment are presented in Table II, and Figure 2 depicts the estimated trajectories in 6 representative sequences. Note that, the proposed R-VIO is lightweight, *without* using any kind of map while the OKVIS does. In general, our R-VIO performs competitively with the OKVIS, and even *better* in some sequences.

### B. Urban Driving dataset

We further conducted tests using a car equipped with our own sensor platform (an Xsens Mti-G unit of IMU/GPS, and

[2]https://github.com/ethz-asl/okvis




(a) V1_01_easy (b) V1_03_difficult (c) V2_02_medium

(d) MH_01_easy (e) MH_03_medium (f) MH_05_difficult

Fig. 2: Trajectory estimates in EuRoC dataset.

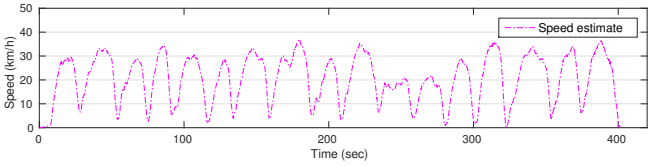
TABLE II: Estimation accuracy (RMSE) in EuRoC dataset

| | OKVIS | | R-VIO | |
|---|---|---|---|---|
| | Orientation (°) | Position (m) | Orientation (°) | Position (m) |
| V1_01_easy | 2.3503 | 0.1420 | **2.1512** | **0.0851** |
| V1_02_medium | 3.3634 | 0.2996 | **0.7778** | **0.1565** |
| V1_03_difficult | 3.5869 | 0.2657 | **0.7295** | **0.1377** |
| V2_01_easy | **0.6511** | 0.3114 | 1.0141 | **0.2161** |
| V2_02_medium | 2.9861 | 0.3414 | **1.2140** | **0.3137** |
| V2_03_difficult | 5.9126 | **0.3771** | **1.2756** | 0.4414 |
| MH_01_easy | **1.0516** | 0.5906 | 1.2367 | **0.3873** |
| MH_02_easy | 1.0624 | **0.6982** | **0.9468** | 0.7406 |
| MH_03_medium | 2.3368 | 0.5504 | **1.3519** | **0.3587** |
| MH_04_difficult | **0.2864** | **0.4312** | 3.5254 | 1.0376 |
| MH_05_difficult | **1.1368** | **0.6743** | 1.3924 | 0.8581 |
| Mean (Seq.1-6) | 3.1417 | 0.2895 | **1.1937** | **0.2250** |
| Mean (Seq.7-11) | **1.1748** | **0.5881** | 1.6906 | 0.6764 |

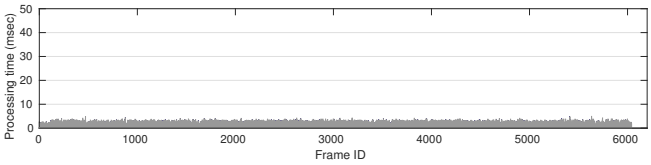
a FLIR Bumblebee2 stereo pair @ 15Hz) and driving on the city streets of Newark, DE. The IMU provided measurements at 400Hz, while the GPS signal was received at 4Hz as the position ground truth (in 2.5∼5.0m precision). Only the left camera images were used for vision inputs, with 200 features being uniformly extracted from each image. It is important to point out that the tests are very challenging primarily due to: (i) traffic lights before which we must stop and wait for 15 to 25 seconds, (ii) frequent yield/stop signs at which we must decelerate or stop, (iii) dynamic scenes including the running vehicles and walking pedestrians in vicinity, (iv) strong lens glare when the camera is facing the sun, and (v) high speeds of vehicle when driving in some areas. Because of these, the OKVIS was not able to provide reasonable localization results, however, the proposed R-VIO performed well. The results are summarized in Table III, and Figure 3 and 4 show the estimated trajectory and speed of each route. Note that, *no* particular treatments, such as zero-velocity update, were used for (i) and (ii). Depending on the driving speed, the sliding window size 10 (for route 1) and 20 (for route 2) were used, and the average processing time of the proposed robocentric EKF is 2.5ms (for route 1) and 4.5ms (for route 2) per frame, which are sufficiently within the realtime requirements. Note also that, as the local gravity effect is jointly estimated, the

(a) Sample images recorded during the driving test.


(b) Estimated trajectory vs. ground truth (GPS track).




(c) Estimated vehicle speed.




(d) Robocentric EKF processing time (sliding-window size 10).

Fig. 3: Route 1: driving in a residential community

TABLE III: Estimation accuracy (RMSE) in driving dataset

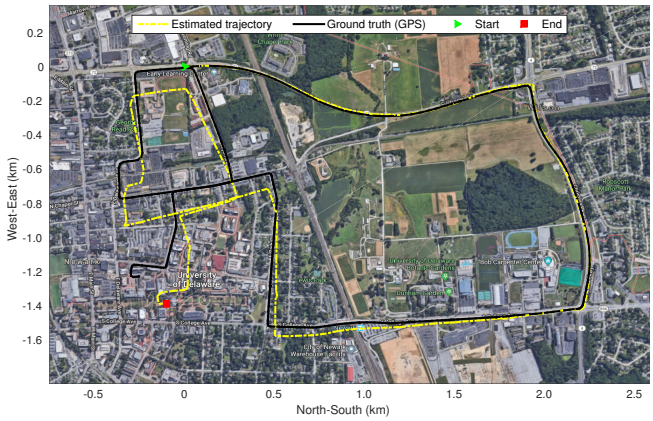
| | Length / Duration | Max. Speed (km/h) | Position RMSE | | |
|---|---|---|---|---|---|
| | | | $x$ (m) | $y$ (m) | $z$ (m) |
| ♯1 | 2.4km / 7min | 36.9 | 7.0426 | 4.9966 | **1.2902** |
| ♯2 | 9.8km / 15min | 85.9 | 30.9340 | 68.5617 | **8.4187** |

$z$-axis drifts are much smaller than the $x$-$y$ position errors. In this challenging driving scenario, without using any kind of map, our R-VIO achieves the average position RMSE of 0.36% and 0.77% of the travelled distance of each route.

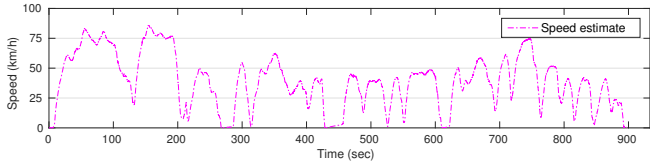
## V. Conclusions and Future Work

In this paper, we have reformulated the VINS with respect to a moving local frame and developed a lightweight, high-precision, robocentric visual-inertial odometry (R-VIO) algorithm. With this novel reformulation, the resulting VINS does not suffer from the observability mismatch issue encountered in the world-centric systems, thus improving the consistency and accuracy. Extensive Monte Carlo simulations and real-world experiments with different sensor platforms navigating

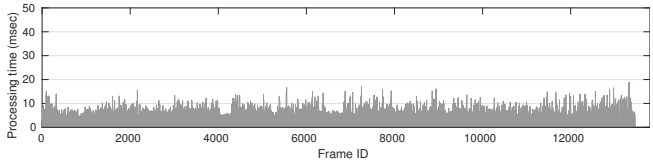
(a) Sample images recorded during the driving test.




(b) Estimated trajectory vs. ground truth (GPS track).




(c) Estimated vehicle speed.




(d) Robocentric EKF processing time (sliding-window size 20).

Fig. 4: Route 2: driving on the suburban and city streets

in different environments and using only monocular vision validate our theoretical analysis and show that the proposed R-VIO is versatile and robust to different types of motions and environments. In future, we will focus on improving the proposed approach further, for example, by integrating online calibration and loop closure to deal with sensor parameter variations and bound localization errors.

## REFERENCES


[1] A. I. Mourikis and S. I. Roumeliotis, "A multi-state constraint kalman filter for vision-aided inertial navigation," in *IEEE Intl. Conf. Robot. Autom. (ICRA)*, 2007, pp. 3565–3572.

[2] E. S. Jones and S. Soatto, "Visual-inertial navigation, mapping and localization: A scalable real-time causal approach," *Intl. J. Robot. Res. (IJRR)*, vol. 30, no. 4, pp. 407–430, 2011.

[3] J. Kelly and G. S. Sukhatme, "Visual-inertial sensor fusion: Localization, mapping and sensor-to-sensor self-calibration," *Intl. J. Robot. Res. (IJRR)*, vol. 30, no. 1, pp. 56–79, 2011.

[4] M. Li and A. I. Mourikis, "High-precision, consistent ekf-based visual-inertial odometry," *Intl. J. Robot. Res. (IJRR)*, vol. 32, no. 6, pp. 690–711, 2013.

[5] S. Leutenegger, S. Lynen, M. Bosse, R. Siegwart, and P. Furgale, "Keyframe-based visual–inertial odometry using nonlinear optimization," *Intl. J. Robot. Res. (IJRR)*, vol. 34, no. 3, pp. 314–334, 2015.

[6] S. Shen, N. Michael, and V. Kumar, "Tightly-coupled monocular visual-inertial fusion for autonomous flight of rotorcraft mavs," in *IEEE Intl. Conf. Robot. Autom. (ICRA)*, 2015, pp. 5303–5310.

[7] V. Usenko, J. Engel, J. Stückler, and D. Cremers, "Direct visual-inertial odometry with stereo cameras," in *IEEE Intl. Conf. Robot. Autom. (ICRA)*, 2016, pp. 1885–1892.

[8] R. Mur-Artal and J. D. Tardós, "Visual-inertial monocular slam with map reuse," *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 796–803, 2017.

[9] G. P. Huang, A. I. Mourikis, and S. I. Roumeliotis, "Observability-based rules for designing consistent ekf slam estimators," *Intl. J. of Robot. Res. (IJRR)*, vol. 29, no. 5, pp. 502–528, 2010.

[10] J. A. Hesch, D. G. Kottas, S. L. Bowman, and S. I. Roumeliotis, "Consistency analysis and improvement of vision-aided inertial navigation," *IEEE Trans. Robot. (TRO)*, vol. 30, no. 1, pp. 158–176, 2014.

[11] G. Huang, M. Kaess, and J. J. Leonard, "Towards consistent visual-inertial navigation," in *IEEE Intl. Conf. Robot. Autom. (ICRA)*, 2014, pp. 4926–4933.

[12] T. Zhang, K. Wu, J. Song, S. Huang, and G. Dissanayake, "Convergence and consistency analysis for a 3-d invariant-ekf slam," *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 733–740, 2017.

[13] J. A. Castellanos, J. Neira, and J. D. Tardós, "Limits to the consistency of ekf-based slam," *IFAC Proceedings Volumes*, vol. 37, no. 8, pp. 716–721, 2004.

[14] J. A. Castellanos, R. Martinez-Cantin, J. D. Tardós, and J. Neira, "Robocentric map joining: Improving the consistency of ekf-slam," *Robot. and Autonom. Sys. (RAS)*, vol. 55, no. 1, pp. 21–29, 2007.

[15] J. Civera, O. G. Grasa, A. J. Davison, and J. Montiel, "1-point ransac for ekf-based structure from motion," in *IEEE/RSJ Intl. Conf. Intell. Robot. Syst. (IROS)*, 2009, pp. 3498–3504.

[16] M. Bloesch, M. Burri, S. Omari, M. Hutter, and R. Siegwart, "Iterated extended kalman filter based visual-inertial odometry using direct photometric feedback," *Intl. J. of Robot. Res. (IJRR)*, vol. 36, no. 10, pp. 1053–1072, 2017.

[17] W. G. Breckenridge, "Quaternions proposed standard conventions," NASA Jet Propulsion Laboratory, Tech. Rep., 1979.

[18] N. Trawny and S. I. Roumeliotis, "Indirect kalman filter for 3d attitude estimation," MARS, University of Minnesota, Tech. Rep., 2005.

[19] Z. Huai and G. Huang, "Robocentric visual-inertial odometry," RPNG, University of Delaware, Tech. Rep., 2018, http://udel.edu/~ghuang/papers/tr_rvio.pdf.

[20] K. Eckenhoff, P. Geneva, and G. Huang, "High-accuracy preintegration for visual-inertial navigation," in *Intl. Workshop on the Algorithmic Foundations of Robotics*, 2016.

[21] J. Civera, A. J. Davison, and J. M. Montiel, "Inverse depth parametrization for monocular slam," *IEEE Trans. Robot. (TRO)*, vol. 24, no. 5, pp. 932–945, 2008.

[22] G. H. Golub and C. F. Van Loan, *Matrix Computations*. JHU Press, 2012, vol. 3.

[23] Z. Chen, K. Jiang, and J. C. Hung, "Local observability matrix and its application to observability analyses," in *the 16th Annual Conf. of IEEE Industrial Electronic Society*, 1990, pp. 100–103.

[24] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, *Estimation with applications to tracking and navigation*. New York: John Wiley & Sons, 2001.

[25] J. Shi *et al.*, "Good features to track," in *IEEE Conf. Comp. Vis. Patt. Reco. (CVPR)*, 1994, pp. 593–600.

[26] S. Baker and I. Matthews, "Lucas-kanade 20 years on: A unifying framework," *Intl. J. Comp. Vis. (IJCV)*, vol. 56, no. 3, pp. 221–255, 2004.

[27] C. Troiani, A. Martinelli, C. Laugier, and D. Scaramuzza, "2-point-based outlier rejection for camera-imu systems with applications to micro aerial vehicles," in *IEEE Intl. Conf. Robot. Autom. (ICRA)*, 2014, pp. 5530–5536.

[28] M. Burri, J. Nikolic, P. Gohl, T. Schneider, J. Rehder, S. Omari, M. W. Achtelik, and R. Siegwart, "The euroc micro aerial vehicle datasets," *Intl. J. Robot. Res. (IJRR)*, vol. 35, no. 10, pp. 1157–1163, 2016.