Square-Root Robocentric Visual-Inertial Odometry With Online Spatiotemporal Calibration

Zheng Huai and Guoquan Huang, *Senior Member, IEEE*

***Abstract*—Robocentric visual-inertial odometry (R-VIO) in our recent work [1] models the probabilistic state estimation problem with respect to a moving local (body) frame, which is contrary to a fixed global (world) frame as in the world-centric formulation, thus avoiding the observability mismatch issue and achieving better estimation consistency. To further improve efficiency and robustness in order to be amenable for the resource-constrained applications, in this paper, we propose a novel information-based estimator, termed R-VIO2. In particular, the numerical stability and computational efficiency are significantly boosted by using i) the square-root expression and ii) incremental QR-based update combined with back substitution. Moreover, the spatial transformation and time offset between visual and inertial sensors are jointly calibrated online to robustify the estimator performance in the presence of unknown parameter errors. The proposed R-VIO2 has been extensively tested on public benchmark dataset as well as in a large-scale real-world experiment, and shown to achieve very competitive accuracy and superior time efficiency against the state-of-the-art visual-inertial navigation methods.**

***Index Terms*—Localization, vision-based navigation.**

I. Introduction and Related Work

VISUAL-INERTIAL odometry (VIO) that typically combines the inertial data from inertial measurement unit (IMU) and the visual observations from camera to compute the orientation and position of the sensing platform has been becoming popular for GPS-denied navigation applications, ranging from the augmented/virtual reality (AR/VR), autonomous driving to even the unmanned planet exploration. Especially, in order for computational efficiency, a VIO is usually realized by either extended Kalman filter (EKF) or fixed-lag smoothing (FLS) which optimizes over a bounded-size sliding window of recent states by marginalizing the past states periodically, for which a complete review of the recent efforts can be found in [2].

Regarding the primary function of VIO, that is, to output the poses of the sensing platform in the (unknown) environments, most proposed approaches solved this problem from a global perspective where fixed, global (world) frame, which is usually aligned with the gravity, is chosen as the navigation reference such that the absolute pose can be estimated with respect to it directly. Therefore, we can term them the *world-centric* VIO. A well known issue for such approaches is the observability mismatch between original and linearized systems which has been identified by different works using linear and nonlinear analyses (e.g., [3], [4]). As a result, many remedies have also been proposed for fixing it (e.g., [5]–[8]), however, most by trading off the accuracy or efficiency of the systems. From another perspective, inspired by the human behaviors in reality, we reformulated the VIO problem in local [1], where the body frame of robot was used as instantaneous navigation frame of reference. Instead, the relative pose between every two locations of robot is estimated, and the current pose with respect to the start (body) frame can always be recovered by incrementally merging new relative pose estimates. Regarding those properties, our approach is termed the *robocentric* VIO. We proved in [1] that the observability mismatch issue does not exist for our proposed robocentric model, thus improving the consistency of VIO estimator fundamentally. Therefore, in this paper, we are going to take such advantage into our new estimator design and further extend its application.

As we summarized in [2], most VIO algorithms are based on EKF or FLS which correspond to the covariance-based or the information-based estimation. Unfortunately, in the sense of *sliding-window* estimation, either covariance or information matrix of the estimator may become dense inevitably because of the marginalization operation. As a result, neither approach has distinguishable computational advantage. To improve this aspect, a practical idea should be to reduce the number of entries involved in the matrix computations. To this end, the square-root expression is employed in our design. We should note that the square-root formulation had been adopted early in [9] to solve a batch estimator. Moreover, based on that the square root information matrix was proposed in [10] with QR factorization-based incremental update scheme that made the proposed approach feasible for real-time application. As only a half-size information matrix is used in the computation, the memory cost is reduced. The condition number for estimator is also square rooted by using the square-root expression, thus improving the numerical stability. Most importantly, QR factorization makes the update incremental as the measurements come in, which bounds the amount of computation needed at every timestep. Therefore, such approach is also very suitable for the resource-constrained applications [11], [12].

It should be noted that, as a multi-sensor system, the VIO performance highly relies on the accuracy of the values of the following parameters including: a) The rotation and translation between camera and IMU, which is known as the camera-IMU extrinsic parameters, and b) A remaining time offset between the respective timestamps of camera and IMU measurements caused by the sensor latency. Although we can calibrate these

Manuscript received 24 February 2022; accepted 15 June 2022. Date of publication 15 July 2022; date of current version 29 July 2022. This letter was recommended for publication by Associate Editor U. Frese and Editor S. Behnke upon evaluation of the reviewers' comments. This work was supported by the University of Delaware College of Engineering and the NSFIIS-1924897. *(Corresponding author: Zheng Huai.)*

The authors are with the Robot Perception and Navigation Group, Department of Mechanical Engineering, University of Delaware, Newark, DE 19716 USA (e-mail: zhuai@udel.edu; ghuang@udel.edu).

Digital Object Identifier 10.1109/LRA.2022.3191209

parameters offline (e.g., [13]), their estimated values may still contain unknown errors, which are relevant to the calibration setups or methods. Beside, the true values of those parameters may be varying in the environment because of the temperature change or platform vibration. Therefore, the ability of refining or calibrating those parameters online becomes crucial in real applications (e.g., [14], [15]). Specifically, in this paper, we will incorporate the information from both IMU and camera measurements for doing online spatiotemporal calibration.

The main contributions of the paper include:

- We derive a novel square-root robocentric formulation for the sliding-window visual-inertial estimation, where online calibration is used to deal with unknown errors in both spatial and temporal sensor calibration parameters. Especially, our proposed estimator is formulated based on a least-squares minimization problem, for which we present in details the cost function formed by the terms derived with our robocentric IMU and camera models.
- We perform extensive evaluations using both challenging benchmark dataset and field sensor data from our large-scale real-world experiment, showing that our developed R-VIO2 achieves very competitive accuracy and superior time efficiency when comparing with the state-of-the-art visual-inertial navigation methods. Especially, to further benefit the research community, we open source our code at https://github.com/rpng/R-VIO2.

II. Square-Root MAP State Estimator

In this section, we briefly introduce the square-root estimator for a maximum-a-posteriori (MAP) estimation problem. Especially, we have a vector of the states to be estimated, $\mathbf{x}$, with the measurements, $\mathcal{Z}$.[1] The prior knowledge about $\mathbf{x}$ is usually modeled by a Gaussian distribution $p(\mathbf{x})$. Then, based on those, the posterior distribution of $\mathbf{x}$ can be expressed as

$$p(\mathbf{x}|\mathcal{Z}) \propto p(\mathbf{x})p(\mathcal{Z}|\mathbf{x}) = p(\mathbf{x}) \prod_{\mathbf{z}_i \in \mathcal{Z}} p(\mathbf{z}_i|\mathbf{x}) \tag{1}$$

where the priori of $\mathbf{x}$ follows $\mathcal{N}(\hat{\mathbf{x}}, \mathbf{\Omega})$, and $\mathbf{z}_i = \mathbf{h}_i(\mathbf{x}) + \mathbf{n}_i$ with $\mathbf{h}_i(\cdot)$ the measurement model and $\mathbf{n}_i$ a zero-mean white Gaussian noise following $\mathcal{N}(\mathbf{0}, \mathbf{\Sigma}_i)$. To find MAP estimate of $\mathbf{x}$ which corresponds to the maximum of $p(\mathbf{x}|\mathcal{Z})$, here we can equivalently minimize the negative logarithm of (1):

$$\begin{aligned}\mathbf{x}^* &= \arg\min_{\mathbf{x}} -\log p(\mathbf{x}|\mathcal{Z}) \\ &= \arg\min_{\mathbf{x}} \|\mathbf{x} - \hat{\mathbf{x}}\|^2_{\mathbf{\Omega}} + \sum_{\mathbf{z}_i \in \mathcal{Z}} \|\mathbf{z}_i - \mathbf{h}_i(\mathbf{x})\|^2_{\mathbf{\Sigma}_i}\end{aligned} \tag{2}$$

where we have employed the notation $\|\mathbf{e}\|^2_{\mathbf{\Lambda}} = \mathbf{e}^\top \mathbf{\Lambda}^{-1}\mathbf{e}$, that is the squared Mahalanobis norm of $\mathbf{e}$ with its covariance $\mathbf{\Lambda}$. To solve this problem which is usually nonlinear because of $\mathbf{h}(\cdot)$, we linearize its cost function at $\hat{\mathbf{x}}$ so as to have

$$\begin{aligned}\mathcal{C}(\hat{\mathbf{x}} + \tilde{\mathbf{x}}) &\simeq \|\tilde{\mathbf{x}}\|^2_{\mathbf{\Omega}} + \sum_{\mathbf{z}_i \in \mathcal{Z}} \|\mathbf{z}_i - \mathbf{h}_i(\hat{\mathbf{x}}) - \mathbf{H}_i\tilde{\mathbf{x}}\|^2_{\mathbf{\Sigma}_i} \\ &= \|\tilde{\mathbf{x}}\|^2_{\mathbf{\Omega}} + \|\mathbf{H}\tilde{\mathbf{x}} - \mathbf{e}\|^2_{\mathbf{\Sigma}}\end{aligned} \tag{3}$$

[1]In what follows, $\hat{\mathbf{x}}$ is used to denote the estimate of $\mathbf{x}$, with $\tilde{\mathbf{x}} \triangleq \mathbf{x} - \hat{\mathbf{x}}$ the corresponding error (or correction) to this estimate. $\mathbf{I}_n$ and $\mathbf{0}_n$ are the $n \times n$ identity and zero matrices, respectively. The left superscript if shown denotes the frame of reference with respect to which a vector is expressed.

where we have stacked all of the Jacobians $\mathbf{H}_i$ and residuals $\mathbf{e}_i = \mathbf{z}_i - \mathbf{h}_i(\hat{\mathbf{x}})$ to have $\mathbf{H}$ and $\mathbf{e}$, with $\mathbf{\Sigma}$ a (block) diagonal matrix of $\mathbf{\Sigma}_i$. As a result, instead of an optimal estimate of $\mathbf{x}$, we find an optimal update to its current estimate $\hat{\mathbf{x}}$, as

$$\tilde{\mathbf{x}}^\oplus = \arg\min_{\tilde{\mathbf{x}}} \mathcal{C}(\hat{\mathbf{x}} + \tilde{\mathbf{x}}) \tag{4}$$

where the superscript $\oplus$ means this solution is optimal up to the linearization errors. It should be noted that, using (upper triangular) Cholesky factors of $\mathbf{\Omega}$ and $\mathbf{\Sigma}$, we can convert (3) into *linear* least-squares form for better numerical stability:

$$\mathcal{C} = \mathcal{C}_{\text{Prior}} + \mathcal{C}_{\text{Measurements}} = \|\mathbf{R}\tilde{\mathbf{x}}\|^2 + \|\mathbf{J}\tilde{\mathbf{x}} - \mathbf{r}\|^2 \tag{5}$$

where $\mathbf{R} = \mathbf{\Omega}^{-1/2}$ (i.e., the *square root information* matrix), while for $\mathbf{J}$ and $\mathbf{r}$ we have $\mathbf{J}_i = \mathbf{\Sigma}_i^{-1/2}\mathbf{H}_i$ and $\mathbf{r}_i = \mathbf{\Sigma}_i^{-1/2}\mathbf{e}_i$ with respect to each measurement, $\mathbf{z}_i$. More importantly, by noticing that $\mathbf{R}$ is upper-triangular, this cost function is able to be updated using QR factorization [16]:

$$\begin{aligned}\mathcal{C} &= \left\| \begin{bmatrix}\mathbf{R} \\ \mathbf{J}\end{bmatrix} \tilde{\mathbf{x}} - \begin{bmatrix}\mathbf{0} \\ \mathbf{r}\end{bmatrix} \right\|^2 = \left\| \mathbf{Q} \begin{bmatrix}\mathbf{R}^\oplus \\ \mathbf{0}\end{bmatrix} \tilde{\mathbf{x}} - \begin{bmatrix}\mathbf{0} \\ \mathbf{r}\end{bmatrix} \right\|^2 \\ &= \left\| \begin{bmatrix}\mathbf{R}^\oplus \\ \mathbf{0}\end{bmatrix} \tilde{\mathbf{x}} - \mathbf{Q}^\top \begin{bmatrix}\mathbf{0} \\ \mathbf{r}\end{bmatrix} \right\|^2 = \left\| \begin{bmatrix}\mathbf{R}^\oplus \\ \mathbf{0}\end{bmatrix} \tilde{\mathbf{x}} - \begin{bmatrix}\mathbf{r}^\oplus \\ \boldsymbol{\epsilon}\end{bmatrix} \right\|^2 \\ &= \left\|\mathbf{R}^\oplus\tilde{\mathbf{x}} - \mathbf{r}^\oplus\right\|^2 + \| - \boldsymbol{\epsilon}\|^2 = \mathcal{C}^\oplus + \|\boldsymbol{\epsilon}\|^2\end{aligned} \tag{6}$$

After discarding $\|\boldsymbol{\epsilon}\|^2$ as it is irrelevant to $\tilde{\mathbf{x}}$, we have

$$\tilde{\mathbf{x}}^\oplus = \arg\min_{\tilde{\mathbf{x}}} \mathcal{C}^\oplus = \mathbf{R}^{\oplus -1}\mathbf{r}^\oplus \tag{7}$$

which can be efficiently found via back substitution, and the state estimate can be updated as: $\hat{\mathbf{x}}^\oplus = \hat{\mathbf{x}} + \tilde{\mathbf{x}}^\oplus$. Especially, the iterations over (3)–(7) can also be used to further reduce the linearization errors. For real applications, usually the sensor measurements are received by the estimator in the order of their timestamps. Therefore, Givens rotations can be used for *in-place* update in (6) to improve real-time performance of the estimator [10], [11]. In this paper, we are going to take those above advantages in our robocentric estimator design.

III. Square-Root Robocentric VIO

In this section, we present our new square-root robocentric VIO algorithm (R-VIO2). Specifically, we explain in details the inertial and visual cost terms derived from the IMU and camera models, respectively, as well as the *a priori* cost term that evolves with the change of navigation frame of reference.

A. State Vector

To model the system more precisely than [1], we use the following vector to describe the state at (image) timestep k:

$$\mathbf{x}_k = \begin{bmatrix}\mathbf{x}_{\mathcal{G}_k}^\top & \mathbf{x}_{\mathcal{P}_k}^\top & \mathbf{x}_{\mathcal{W}_k}^\top\end{bmatrix}^\top \tag{8}$$

where $\mathbf{x}_{\mathcal{G}_k} = [{}^k_G\bar{q}^\top \ {}^{R_k}\mathbf{p}_G^\top \ {}^{R_k}\mathbf{g}^\top]^\top$ is the global state with ${}^k_G\bar{q}$ (4×1 unit quaternion [17]) and ${}^{R_k}\mathbf{p}_G$ (3×1 Euclidean coordinate) the orientation and position of global frame $\{G\}$ with respect to the robocentric frame of reference $\{R_k\}$ (i.e., coincident with IMU frame $\{I\}$ at timestamp t_k) and ${}^{R_k}\mathbf{g}$ a unit vector of local gravity in $\{R_k\}$; $\mathbf{x}_{\mathcal{P}_k} = [{}^C_I\bar{q}^\top \ {}^C\mathbf{p}_I^\top \ t_d]^\top$ comprises of online calibration parameters with ${}^C_I\bar{q}$ and ${}^C\mathbf{p}_I$

the rotation and translation between camera frame $\{C\}$ and IMU frame $\{I\}$, and t_d the difference between the receiving timestamps of camera and IMU measurements at timestep k (e.g., the IMU measurement paired with image k should be the one at timestamp $t_k + t_d$); and both $\mathbf{x}_{\mathcal{G}_k}$ and $\mathbf{x}_{\mathcal{P}_k}$ are of zero kinematics. Specifically, $\mathbf{x}_{\mathcal{W}_k}$ is a sliding window of the relative IMU poses *between* recent N timesteps including k:

$$\mathbf{x}_{\mathcal{W}_k} = \begin{bmatrix}\mathbf{x}_{w_1}^\top & \cdots & \mathbf{x}_{w_{N-2}}^\top & \mathbf{x}_{w_{N-1}}^\top\end{bmatrix}^\top \tag{9}$$

$$\mathbf{x}_{w_{1:N-2}} = \begin{bmatrix}{}^{i}_{i-1}\bar{q}^\top & {}^{R_{i-1}}\mathbf{p}_{R_i}^\top\end{bmatrix}^\top \text{ for } i = k-N+2 : k-1,$$

$$\mathbf{x}_{w_{N-1}} = \begin{bmatrix}{}^{k}_{k-1}\bar{q}^\top & {}^{R_{k-1}}\mathbf{p}_{R_k}^\top & {}^{I_k}\mathbf{v}^\top & \mathbf{b}_{g_k}^\top & \mathbf{b}_{a_k}^\top\end{bmatrix}^\top \tag{10}$$

where $\mathbf{x}_{w_n}$ saves the orientation and position of $\{R_n\}$ with respect to $\{R_{n-1}\}$, while $\mathbf{x}_{w_{N-1}}$ additionally keeps the IMU velocity ${}^{I_k}\mathbf{v}$ expressed in $\{I_k\}$, and biases, $\mathbf{b}_{g_k}$ and $\mathbf{b}_{a_k}$.

B. IMU Term

We propose a preintegrated IMU cost term, for which our robocentric motion model proposed in [1] is used to process IMU measurements between image k and $k+1$. Especially, the relative motion estimate from $\{R_k\}$ to $\{R_{k+1}\}$, $\hat{\mathbf{x}}_{w_N} = [{}^{k+1}_{k}\hat{\bar{q}}^\top \ {}^{R_k}\hat{\mathbf{p}}_{R_{k+1}}^\top \ {}^{I_{k+1}}\hat{\mathbf{v}}^\top \ \hat{\mathbf{b}}_{g_{k+1}}^\top \ \hat{\mathbf{b}}_{a_{k+1}}^\top]^\top$, is computed and appended to the sliding window in $\hat{\mathbf{x}}_k$, so that we have the estimate of intermediate state: $\hat{\boldsymbol{\chi}}_{k+1} = [\hat{\mathbf{x}}_{\mathcal{G}_k}^\top \ \hat{\mathbf{x}}_{\mathcal{P}_k}^\top \ \hat{\mathbf{x}}_{\mathcal{W}_{k+1}}^\top]^\top$. To incorporate the corresponding measurement information, we construct a minimal state vector $\mathbf{y}_k = [\mathbf{x}_{\mathcal{G}_k}^\top \ \mathbf{x}_{\mathcal{I}_k}^\top]^\top$ with $\mathbf{x}_{\mathcal{I}_k} = [{}^{k}_{k}\bar{q}^\top \ {}^{R_k}\mathbf{p}_{I_k}^\top \ {}^{I_k}\mathbf{v}^\top \ \mathbf{b}_{g_k}^\top \ \mathbf{b}_{a_k}^\top]^\top$, which includes all the states that are used in the IMU propagation. Referring to [18], the error-state transition matrix $\boldsymbol{\Phi}_{k+1,k}$ can be computed by using the IMU measurements in $[t_k + t_d, t_{k+1} + t_d]$, aligned with the corresponding image time interval, such that

$$\boldsymbol{\Phi}_{k+1,k} = \prod_{\tau \in [t_k+t_d,\, t_{k+1}+t_d]} \boldsymbol{\Phi}_{\tau+1,\tau} \tag{11}$$

Based on that, we have $\tilde{\mathbf{y}}_{k+1} = \boldsymbol{\Phi}_{k+1,k}\tilde{\mathbf{y}}_k + \mathbf{n}_{k+1,k}$, where $\mathbf{y}_{k+1} = [\mathbf{x}_{\mathcal{G}_k}^\top \ \mathbf{x}_{w_N}^\top]^\top$ and $\mathbf{n}_{k+1,k}$ is the preintegrated noise vector, which can further be expressed in details as:[2]

$$\begin{bmatrix}\tilde{\mathbf{x}}_{\mathcal{G}_k} \\ \tilde{\mathbf{x}}_{w_N}\end{bmatrix} = \begin{bmatrix}\mathbf{I}_9 & \mathbf{0}_{9\times 15} \\ \boldsymbol{\Phi}_{\mathcal{G}} & \boldsymbol{\Phi}_{\mathcal{I}}\end{bmatrix}\begin{bmatrix}\tilde{\mathbf{x}}_{\mathcal{G}_k} \\ \tilde{\mathbf{x}}_{\mathcal{I}_k}\end{bmatrix} + \begin{bmatrix}\mathbf{0}_{9\times 1} \\ \mathbf{n}_{\mathcal{I}}\end{bmatrix} \tag{12}$$

$$\boldsymbol{\Phi}_{\mathcal{G}} = \begin{bmatrix}\mathbf{0}_{15\times 3} & \mathbf{0}_{15\times 3} & \boldsymbol{\Phi}_{\mathbf{g}}\end{bmatrix},$$

$$\boldsymbol{\Phi}_{\mathcal{I}} = \begin{bmatrix}\boldsymbol{\Phi}_{\boldsymbol{\theta}} & \boldsymbol{\Phi}_{\mathbf{p}} & \boldsymbol{\Phi}_{\mathbf{v}} & \boldsymbol{\Phi}_{\mathbf{b}_g} & \boldsymbol{\Phi}_{\mathbf{b}_a}\end{bmatrix} \tag{13}$$

We should note that $\mathbf{x}_{\mathcal{I}_k}$ does not exist in the actual state vector $\boldsymbol{\chi}_{k+1}$, thus (12) has not been a valid relationship yet. However, by further examining the entries in $\tilde{\mathbf{x}}_{\mathcal{I}_k}$ we should immediately find that $\delta^k_k\boldsymbol{\theta} \equiv \mathbf{0}_{3\times 1}$ and ${}^{R_k}\tilde{\mathbf{p}}_{I_k} \equiv \mathbf{0}_{3\times 1}$. This allows us to replace $\tilde{\mathbf{x}}_{\mathcal{I}_k}$ with $\tilde{\mathbf{x}}_{w_{N-1}}$ that exists in the error state $\tilde{\boldsymbol{\chi}}_{k+1}$. Therefore, a valid relationship can be derived as

$$\tilde{\mathbf{x}}_{w_N} = \mathbf{H}_{\mathcal{G}}\tilde{\mathbf{x}}_{\mathcal{G}_k} + \mathbf{H}_{w_{N-1}}\tilde{\mathbf{x}}_{w_{N-1}} + \mathbf{n}_{\mathcal{I}} \tag{14}$$

$$\mathbf{H}_{\mathcal{G}} = \boldsymbol{\Phi}_{\mathcal{G}},$$

$$\mathbf{H}_{w_{N-1}} = \begin{bmatrix}\mathbf{0}_{15\times 3} & \mathbf{0}_{15\times 3} & \boldsymbol{\Phi}_{\mathbf{v}} & \boldsymbol{\Phi}_{\mathbf{b}_g} & \boldsymbol{\Phi}_{\mathbf{b}_a}\end{bmatrix} \tag{15}$$

[2]For the error quaternion we quantify it using its associated 3-dimension error angle $\delta\boldsymbol{\theta}$ (i.e., $\delta\bar{q} \simeq [\frac{1}{2}\delta\boldsymbol{\theta}^\top \ 1]^\top$), therefore, we use $\boldsymbol{\Phi}_{\boldsymbol{\theta}}$ not $\boldsymbol{\Phi}_{\bar{q}}$ [19].

Accordingly, the residual of our IMU cost term is given by

$$\mathbf{e}_I = \mathbf{H}_I\tilde{\boldsymbol{\chi}}_{k+1} + \mathbf{n}_{\mathcal{I}} \tag{16}$$

$$\mathbf{H}_I = \begin{bmatrix}\mathbf{H}_{\mathcal{G}} & \mathbf{0}_{15\times 7} & [\mathbf{0}_{15\times 6(N-2)} & \mathbf{H}_{w_{N-1}} & -\mathbf{I}_{15}]\end{bmatrix} \tag{17}$$

Next, to get the associated covariance of the preintegrated noise $\mathbf{n}_{k+1,k}$, we perform covariance propagation along with (11), however, starting from the initial value $\boldsymbol{\Sigma}_{k,k} = \mathbf{0}_{24}$:

$$\boldsymbol{\Sigma}_{\tau+1,k} = \boldsymbol{\Phi}_{\tau+1,\tau}\boldsymbol{\Sigma}_{\tau,k}\boldsymbol{\Phi}_{\tau+1,\tau}^\top + \mathbf{G}\mathbf{Q}\mathbf{G}^\top \tag{18}$$

where $\mathbf{G}$ and $\mathbf{Q}$ are the IMU noise Jacobian and covariance matrices, respectively [1]. It is easy to verify that $\boldsymbol{\Sigma}_{k+1,k}$ is in the following form with $\boldsymbol{\Sigma}_{\mathcal{I}}$ the covariance matrix of $\mathbf{n}_{\mathcal{I}}$:

$$\boldsymbol{\Sigma}_{k+1,k} = \begin{bmatrix}\mathbf{0}_9 & \mathbf{0}_{9\times 15} \\ \mathbf{0}_{15\times 9} & \boldsymbol{\Sigma}_{\mathcal{I}}\end{bmatrix}. \tag{19}$$

Now, with (16) and $\boldsymbol{\Sigma}_{\mathcal{I}}$, we can give out the square-root expression of our preintegrated IMU cost term as

$$\mathcal{C}_{\text{IMU}} = \|\mathbf{J}_I\tilde{\boldsymbol{\chi}}_{k+1} - \mathbf{r}_I\|^2 \tag{20}$$

$$\mathbf{J}_I = \boldsymbol{\Sigma}_{\mathcal{I}}^{-1/2}\mathbf{H}_I, \quad \mathbf{r}_I = \boldsymbol{\Sigma}_{\mathcal{I}}^{-1/2}\mathbf{e}_I = \mathbf{0}_{15\times 1} \tag{21}$$

C. Camera Term

To build a relationship between the camera measurements and the current state (e.g., $\boldsymbol{\chi}_{k+1}$), let us investigate the case where a landmark L is observed from a series of images $\mathcal{S}$ (e.g., $\{1, \ldots, N+1\}$). Following the perspective projection model, the i-th ($i \in \mathcal{S}$) measurement of L can be given by

$$\mathbf{z}_i = \frac{1}{z_i(t+t_d)}\begin{bmatrix}x_i(t+t_d) \\ y_i(t+t_d)\end{bmatrix} + \mathbf{n}_i \tag{22}$$

$${}^{C_i}\mathbf{p}_L = \begin{bmatrix}x_i(t+t_d) & y_i(t+t_d) & z_i(t+t_d)\end{bmatrix}^\top \tag{23}$$

where $\mathbf{n}_i \sim \mathcal{N}(\mathbf{0}_{2\times 1}, \sigma_{im}^2\mathbf{I}_2)$ is the image noise, and ${}^{C_i}\mathbf{p}_L$ is the position of landmark in the corresponding camera frame $\{C_i\}$, while here we explicitly align it with the actual pose of camera at image timestamp t by taking into account t_d.

Note that, if we know the position of L in its first camera frame, ${}^{C_1}\mathbf{p}_L$, and the rotation and translation between $\{C_1\}$ and $\{C_i\}$, ${}^{i}_{1}\mathbf{C}$ and ${}^{C_i}\mathbf{p}_{C_1}$, then ${}^{C_i}\mathbf{p}_L$ can be given by

$${}^{C_i}\mathbf{p}_L = {}^{i}_{1}\mathbf{C}\,{}^{C_1}\mathbf{p}_L + {}^{C_i}\mathbf{p}_{C_1} \tag{24}$$

for which we can use relative poses in $\mathbf{x}_{\mathcal{W}}$ to compute[3]

$${}^{i}_{1}\mathbf{C} = {}^{C}_{I}\mathbf{C}\,{}^{i}_{\bar{q}1}\mathbf{C}\,{}^{C}_{\bar{q}I}\mathbf{C}\,{}^{\top}_{\bar{q}} \tag{25}$$

$${}^{C_i}\mathbf{p}_{C_1} = {}^{C}_{I}\mathbf{C}_{\bar{q}}\,{}^{R_i}\mathbf{p}_{R_1} + (\mathbf{I}_3 - {}^{i}_{1}\mathbf{C})\,{}^{C}\mathbf{p}_I \tag{26}$$

In particular, we express ${}^{C_1}\mathbf{p}_L$ in an inverse-depth form [20]:

$${}^{C_1}\mathbf{p}_L = \frac{1}{\rho(t+t_d)}\mathbf{u}(\phi(t+t_d), \psi(t+t_d)) \tag{27}$$

$$\mathbf{u} = \begin{bmatrix}\cos\phi(t+t_d)\sin\psi(t+t_d) \\ \sin\phi(t+t_d) \\ \cos\phi(t+t_d)\cos\psi(t+t_d)\end{bmatrix} \tag{28}$$

where $\mathbf{u}$ is the directional vector of ${}^{C_1}\mathbf{p}_L$ with ϕ and ψ the elevation and azimuth angles in $\{C_1\}$, respectively, while ρ

[3]${}^{b}_{a}\mathbf{C}_{\bar{q}} \triangleq \mathbf{C}({}^{b}_{a}\bar{q})$ is the 3×3 rotation matrix derived from quaternion ${}^{b}_{a}\bar{q}$.

is the inverse of depth along $\mathbf{u}$. We denote $\boldsymbol{\lambda} := [\phi, \psi, \rho]^\top$. ${}^{C_1}\mathbf{p}_L$ is correlated with t_d because it is anchored on $\{C_1\}$. By substituting (27) into (24) and normalize it by ρ, we have

$$\rho^{C_i}\mathbf{p}_L = {}^{i}_{1}\mathbf{C}\mathbf{u} + \rho^{C_i}\mathbf{p}_{C_1} =: \mathbf{h}_i(\mathbf{x}_\mathcal{P}, \mathbf{x}_\mathcal{W}, \boldsymbol{\lambda}) \tag{29}$$

and our inverse-depth measurement model is given by

$$\mathbf{z}_i = \frac{1}{h_{i,3}}\begin{bmatrix} h_{i,1} \\ h_{i,2} \end{bmatrix} + \mathbf{n}_i, \quad i \in \mathcal{S} \tag{30}$$

The initial value recovery for $\boldsymbol{\lambda}$ is as in [1], and after that we linearize (30) at $\hat{\boldsymbol{\chi}}$ and $\hat{\boldsymbol{\lambda}}$ for the measurement residual:

$$\mathbf{e}_i = \mathbf{z}_i - \hat{\mathbf{z}}_i \simeq \mathbf{H}_{i,\mathcal{P}}\tilde{\mathbf{x}}_\mathcal{P} + \mathbf{H}_{i,\mathcal{W}}\tilde{\mathbf{x}}_\mathcal{W} + \mathbf{H}_{i,\boldsymbol{\lambda}}\tilde{\boldsymbol{\lambda}} + \mathbf{n}_i \tag{31}$$

(see Appendix for the derivations of $\mathbf{H}_{i,\mathcal{P}}$, $\mathbf{H}_{i,\mathcal{W}}$ and $\mathbf{H}_{i,\boldsymbol{\lambda}}$). By assuming all the measurements are independent, we have the corresponding cost function of L:

$$\mathcal{C}_L = \|\mathbf{H}_{\boldsymbol{\chi}}\tilde{\boldsymbol{\chi}} + \mathbf{H}_{\boldsymbol{\lambda}}\tilde{\boldsymbol{\lambda}} - \mathbf{e}_L\|^2_{\boldsymbol{\Sigma}_L} \tag{32}$$

$$\mathbf{H}_{\boldsymbol{\chi}} = \begin{bmatrix}\mathbf{0}_{2M\times 9} & \mathbf{H}_\mathcal{P} & [\mathbf{H}_{w_{1:N-1}} & \mathbf{0}_{2M\times 9} & \mathbf{H}_{w_N} & \mathbf{0}_{2M\times 9}]\end{bmatrix},$$

$$\mathbf{e}_L = \mathbf{z} - \hat{\mathbf{z}} \tag{33}$$

where we have stacked $\mathbf{H}_{i,\mathcal{P}}$, $\mathbf{H}_{i,\mathcal{W}}$, $\mathbf{H}_{i,\boldsymbol{\lambda}}$, and $\mathbf{e}_i$ of L for $i = 1, \ldots, M$ ($M = |\mathcal{S}|$) to get $\mathbf{H}_{\boldsymbol{\chi}}$, $\mathbf{H}_{\boldsymbol{\lambda}}$ and $\mathbf{e}_L$, with the stacked covariance matrix $\boldsymbol{\Sigma}_L = \sigma^2_{im}\mathbf{I}_{2M}$.

Note that $\boldsymbol{\lambda}$ does not exist in the state vector. Therefore, to derive a valid cost term of L, we must eliminate $\tilde{\boldsymbol{\lambda}}$ from (32). Note also that, in the general case $M \geq 2$, and hence $\mathbf{H}_{\boldsymbol{\lambda}}$ is a tall matrix whose QR decomposition can be given by

$$\mathbf{H}_{\boldsymbol{\lambda}} = \begin{bmatrix}\mathbf{Q}_1 & \mathbf{Q}_2\end{bmatrix}\begin{bmatrix}\mathbf{R}_{\boldsymbol{\lambda}} \\ \mathbf{0}_{(2M-3)\times 3}\end{bmatrix} \tag{34}$$

where $\mathbf{Q}_1$ and $\mathbf{Q}_2$ correspond to the range and nullspace of $\mathbf{H}_{\boldsymbol{\lambda}}$, respectively, which we can use to refactor (32), so that

$$\begin{aligned}\mathcal{C}_L &= \left\| \begin{bmatrix}\mathbf{Q}_1^\top \\ \mathbf{Q}_2^\top\end{bmatrix}\mathbf{H}_{\boldsymbol{\chi}}\tilde{\boldsymbol{\chi}} + \begin{bmatrix}\mathbf{R}_{\boldsymbol{\lambda}} \\ \mathbf{0}\end{bmatrix}\tilde{\boldsymbol{\lambda}} - \begin{bmatrix}\mathbf{Q}_1^\top \\ \mathbf{Q}_2^\top\end{bmatrix}\mathbf{e}_L \right\|^2_{\boldsymbol{\Sigma}_L} \\ &= \|\mathbf{Q}_1^\top\mathbf{H}_{\boldsymbol{\chi}}\tilde{\boldsymbol{\chi}} + \mathbf{R}_{\boldsymbol{\lambda}}\tilde{\boldsymbol{\lambda}} - \mathbf{Q}_1^\top\mathbf{e}_L\|^2_{\boldsymbol{\Sigma}_{L,1}} \\ &\quad + \|\mathbf{Q}_2^\top\mathbf{H}_{\boldsymbol{\chi}}\tilde{\boldsymbol{\chi}} - \mathbf{Q}_2^\top\mathbf{e}_L\|^2_{\boldsymbol{\Sigma}_{L,2}}\end{aligned} \tag{35}$$

where $\boldsymbol{\Sigma}_{L,1} = \sigma^2_{im}\mathbf{I}_3$ and $\boldsymbol{\Sigma}_{L,2} = \sigma^2_{im}\mathbf{I}_{2M-3}$. As a result, we extract the second component of $\mathcal{C}_L$ as camera cost term for L, and convert it into the square-root form:

$$\mathcal{C}_{\text{Camera}} = \|\mathbf{J}_C\tilde{\boldsymbol{\chi}} - \mathbf{r}_C\|^2 \tag{36}$$

$$\mathbf{J}_C = \sigma^{-1}_{im}\mathbf{Q}_2^\top\mathbf{H}_{\boldsymbol{\chi}}, \quad \mathbf{r}_C = \sigma^{-1}_{im}\mathbf{Q}_2^\top\mathbf{e}_L \tag{37}$$

In practice, if more than one landmark are observed, we will stack their $\mathcal{C}_{\text{Camera}}$'s together into one single cost term.

Interestingly, we can also readily figure out the degenerate cases of online calibration by investigating the expression of Jacobian $\mathbf{H}_{i,\mathcal{P}}$. Especially, the zero column(s) in the Jacobian matrix suggests that the sensor measurement(s) do not convey any information about the corresponding state(s) [21]. Bearing this in mind and noting that the primary factors in the Jacobian expression are all related to the sensor motion, we find these cases that can cause zero column(s) in $\mathbf{H}_\mathcal{P}$:

C1) No rotation (i.e., ${}^{I_{1:i}}\hat{\boldsymbol{\omega}} = \mathbf{0}_{3i\times 1} \Rightarrow {}^{i}_{1}\hat{\mathbf{C}} = \mathbf{I}_3$)

C2) No translation (i.e., ${}^{I_{1:i}}\hat{\mathbf{v}} = \mathbf{0}_{3i\times 1} \Rightarrow {}^{C_1}\hat{\mathbf{p}}_{C_i} = \mathbf{0}_{3\times 1}$)[4]

C3) Camera single-axis rotation (i.e., one of the columns in $\mathbf{I}_3 - {}^{i}_{1}\hat{\mathbf{C}}$ becomes $\mathbf{0}_{3\times 1}$)

where ${}^{I_{1:i}}\hat{\boldsymbol{\omega}}$ and ${}^{I_{1:i}}\hat{\mathbf{v}}$ are the estimates of angular and linear velocities of IMU, respectively. Thus, to calibrate both spatial parameters, ${}^{C}_{I}\boldsymbol{\theta}$ and ${}^{C}\mathbf{p}_I$, online, this, in general, requires *at least* 4 degree-of-freedom (DOF) sensor motion where 3DOF rotation is used to avoid C1 and C3, while 1DOF translation is used to avoid C2 and recover the value of ρ. However, t_d is calibratable as long as the sensor platform is in motion, that is, ${}^{I}\hat{\boldsymbol{\omega}} \neq \mathbf{0}$ or ${}^{I}\hat{\mathbf{v}} \neq \mathbf{0}$, as we will show in Section IV.

D. A Priori Term

In the MAP estimation, this term conveys the information of state up to the current time. Especially, at timestep $k+1$ the prior information is given by

$$\mathcal{C}_{\text{Prior},k+1} = \|\mathbf{R}_k\tilde{\mathbf{x}}_k\|^2 = \|\boldsymbol{R}_{k+1}\tilde{\boldsymbol{\chi}}_{k+1}\|^2 \tag{38}$$

where $\boldsymbol{R}_{k+1} = [\mathbf{R}_k \ \ \mathbf{0}_{\gamma\times 15}]$ with γ the dimension of $\mathbf{R}_k$. An optimal update of $\hat{\boldsymbol{\chi}}_{k+1}$ can be solved by following (6)–(7):

$$\mathcal{C}_{k+1} = \mathcal{C}_{\text{Prior},k+1} + \mathcal{C}_{\text{IMU},k+1} + \mathcal{C}_{\text{Camera},k+1} \tag{39}$$

$$\tilde{\boldsymbol{\chi}}^{\oplus}_{k+1} = \arg\min_{\tilde{\boldsymbol{\chi}}_{k+1}} \mathcal{C}_{k+1} \tag{40}$$

Thus, $\hat{\boldsymbol{\chi}}_{k+1}$ is updated as:[5]

$$\hat{\boldsymbol{\chi}}^{\oplus}_{k+1} = \hat{\boldsymbol{\chi}}_{k+1} \boxplus \tilde{\boldsymbol{\chi}}^{\oplus}_{k+1} \tag{41}$$

1) Composition: The changing of the navigation frame of reference is the most *distinguishing* feature of our robocentric VIO [1]. Here, once finishing the update, we shift the frame of reference of $\boldsymbol{\chi}_{k+1}$ from $\{R_k\}$ to $\{R_{k+1}\}$ (i.e., $\{I_{k+1}\}$).

The state vector with respect to $\{R_{k+1}\}$ is $\mathbf{x}_{k+1}$. To have $\hat{\mathbf{x}}_{k+1}$, we need to convert $\hat{\mathbf{x}}^{\oplus}_{\mathcal{G}_k}$ of $\hat{\boldsymbol{\chi}}^{\oplus}_{k+1}$ to $\hat{\mathbf{x}}_{\mathcal{G}_{k+1}}$, which can be done by the following state composition with $\hat{\mathbf{x}}^{\oplus}_{w_N}$:

$${}^{k+1}_{G}\hat{q} = {}^{k+1}_{k}\hat{q}^{\oplus} \otimes {}^{k}_{G}\hat{q}^{\oplus} \tag{42}$$

$${}^{R_{k+1}}\hat{\mathbf{p}}_G = {}^{k+1}_{k}\mathbf{C}^{\oplus}_{\hat{q}}({}^{R_k}\hat{\mathbf{p}}^{\oplus}_G - {}^{R_k}\hat{\mathbf{p}}^{\oplus}_{R_{k+1}}) \tag{43}$$

$${}^{R_{k+1}}\hat{\mathbf{g}} = {}^{k+1}_{k}\mathbf{C}^{\oplus}_{\hat{q}}\,{}^{R_k}\hat{\mathbf{g}}^{\oplus} \tag{44}$$

Note that, such change of frame of reference does not affect $\mathbf{x}_\mathcal{P}$ and $\mathbf{x}_\mathcal{W}$, that is, $\hat{\mathbf{x}}_{\mathcal{P}_{k+1}} = \hat{\mathbf{x}}^{\oplus}_{\mathcal{P}_k}$ and $\hat{\mathbf{x}}_{\mathcal{W}_{k+1}} = \hat{\mathbf{x}}^{\oplus}_{\mathcal{W}_{k+1}}$. In another hand, we should note that after updating $\hat{\boldsymbol{\chi}}_{k+1}$:

$$\mathcal{C}^{\oplus}_{k+1} = \|\boldsymbol{R}^{\oplus}_{k+1}\tilde{\boldsymbol{\chi}}_{k+1}\|^2 \tag{45}$$

where the dimension of $\boldsymbol{R}^{\oplus}_{k+1}$ is $\gamma + 15$. However, the square root information matrix corresponding to $\hat{\mathbf{x}}_{k+1}$ is $\mathbf{R}_{k+1}$, for which we transform $\boldsymbol{R}^{\oplus}_{k+1}$ according to the following lemma:

Lemma 1: Given the Jacobian matrix $\mathbf{V} = \frac{\partial\tilde{\mathbf{x}}}{\partial\tilde{\boldsymbol{\chi}}}$, the (block upper-triangular) square root information matrices of $\mathbf{x}$ and $\boldsymbol{\chi}$, $\mathbf{R}$ and $\boldsymbol{R}$ (of the same size), satisfy $\mathbf{R} = \boldsymbol{R}\mathbf{V}^{-1}$.

Proof: See Appendix. ■

[4]Note that this condition suggests zero parallax between the images such that the estimate of ρ may not be able to converge (i.e., $\hat{\rho} \to 0$) with single camera, and therefore will also have zero column(s) shown in $\mathbf{H}_\mathcal{P}$.

[5]As the updating on quaternion is non-Euclidean (with the multiplication operator $\otimes$ [19]), we use $\boxplus$ to represent a comprehensive state update.

Using Lemma 1, with the Jacobian matrix $\mathbf{V} = \frac{\partial \tilde{\mathbf{x}}_{k+1}}{\partial \tilde{\boldsymbol{\chi}}_{k+1}}$:

$$\mathbf{V} = \begin{bmatrix} \mathbf{V}_{\mathcal{G}} & \mathbf{0}_{9\times 7+6(N-1)} & \mathbf{V}_{w_N} \\ & \mathbf{I}_{7+6(N-1)} & \mathbf{0}_{7+6(N-1)\times 15} \\ & & \mathbf{I}_{15} \end{bmatrix} \tag{46}$$

$$\mathbf{V}_{\mathcal{G}} = \frac{\partial \tilde{\mathbf{x}}_{\mathcal{G}_{k+1}}}{\partial \tilde{\mathbf{x}}_{\mathcal{G}_k}} = \begin{bmatrix} {}^{k+1}_{k}\mathbf{C}_{\hat{\bar{q}}} & \mathbf{0}_3 & \mathbf{0}_3 \\ & {}^{k+1}_{k}\mathbf{C}_{\hat{\bar{q}}} & \mathbf{0}_3 \\ & & {}^{k+1}_{k}\mathbf{C}_{\hat{\bar{q}}} \end{bmatrix} \tag{47}$$

$$\mathbf{V}_{w_N} = \frac{\partial \tilde{\mathbf{x}}_{\mathcal{G}_{k+1}}}{\partial \tilde{\mathbf{x}}_{w_N}} = \begin{bmatrix} \mathbf{I}_3 & \mathbf{0}_3 & \mathbf{0}_{3\times 9} \\ \lfloor {}^{R_{k+1}}\hat{\mathbf{p}}_G \rfloor & -{}^{k+1}_{k}\mathbf{C}_{\hat{\bar{q}}} & \mathbf{0}_{3\times 9} \\ \lfloor {}^{R_{k+1}}\hat{\mathbf{g}} \rfloor & \mathbf{0}_3 & \mathbf{0}_{3\times 9} \end{bmatrix} \tag{48}$$

we have $\mathbf{R}_{k+1} = \boldsymbol{R}^{\oplus}_{k+1}\mathbf{V}^{-1}$, that is

$$\begin{aligned} \mathbf{R}_{k+1} &= \begin{bmatrix} \boldsymbol{R}^{\oplus}_{\mathcal{G}} & \cdots & \boldsymbol{R}^{\oplus}_{\mathcal{G}w_N} \\ & \ddots & \vdots \\ & & \boldsymbol{R}^{\oplus}_{w_N} \end{bmatrix} \begin{bmatrix} \mathbf{V}^{-1}_{\mathcal{G}} & \cdots & -\mathbf{V}^{-1}_{\mathcal{G}}\mathbf{V}_{w_N} \\ & \ddots & \vdots \\ & & \mathbf{I} \end{bmatrix} \\ &= \begin{bmatrix} \boldsymbol{R}^{\oplus}_{\mathcal{G}}\mathbf{V}^{-1}_{\mathcal{G}} & \cdots & \boldsymbol{R}^{\oplus}_{\mathcal{G}w_N} - \boldsymbol{R}^{\oplus}_{\mathcal{G}}\mathbf{V}^{-1}_{\mathcal{G}}\mathbf{V}_{w_N} \\ & \ddots & \vdots \\ & & \boldsymbol{R}^{\oplus}_{w_N} \end{bmatrix} \end{aligned} \tag{49}$$

where $\mathbf{V}^{-1}_{\mathcal{G}} = \mathbf{V}^{\top}_{\mathcal{G}}$, and only the top-left and top-right blocks in $\boldsymbol{R}$ corresponding to $\mathbf{x}_{\mathcal{G}}$ and $\mathbf{x}_{w_N}$, respectively, are altered.

2) Marginalization: To keep the length of sliding window constant, we need to marginalize: a) the oldest relative pose $\mathbf{x}_{w_1}$, and b) the IMU velocity ${}^{I_k}\mathbf{v}$ and biases $\mathbf{b}_{g_k}$ and $\mathbf{b}_{a_k}$ in $\mathbf{x}_{w_{N-1}}$, from $\mathbf{x}_{\mathcal{W}_{k+1}}$. To this end, we have another lemma:

Lemma 2: Given the state vector $\mathbf{x} = [\mathbf{x}^{\top}_m \ \mathbf{x}^{\top}_r]^{\top}$, with its square-root information factor $\mathcal{C} = \|\mathbf{R}\tilde{\mathbf{x}} - \mathbf{r}\|^2$, where

$$\mathbf{R} = \begin{bmatrix} \mathbf{R}_{mm} & \mathbf{R}_{mr} \\ & \mathbf{R}_{rr} \end{bmatrix}, \quad \mathbf{r} = \begin{bmatrix} \mathbf{r}_m \\ \mathbf{r}_r \end{bmatrix} \tag{50}$$

a complete information factor of $\mathbf{x}_r$ only is $\|\mathbf{R}_{rr}\tilde{\mathbf{x}}_r - \mathbf{r}_r\|^2$.

Proof: See Appendix. ■

According to Lemma 2, we put the states to be marginalized into $\mathbf{x}_m$ while grouping the remaining ones in $\mathbf{x}_r$, such that $\hat{\mathbf{x}}^{\ominus}_{k+1} = [\hat{\mathbf{x}}^{\top}_m \ \hat{\mathbf{x}}^{\top}_r]^{\top}$ ($\ominus$ means reordered). Meanwhile, the columns in $\mathbf{R}_{k+1}$ are reordered according to the state order in $\hat{\mathbf{x}}^{\ominus}_{k+1}$, which gives us $\mathbf{R}^{\ominus}_{k+1}$. Next, a QR factorization is performed to make $\mathbf{R}^{\ominus}_{k+1}$ upper-triangular again. Finally, we finish marginalization with $\hat{\mathbf{x}}_{k+1} = \hat{\mathbf{x}}_r$ and $\mathbf{R}_{k+1} = \mathbf{R}_{rr}$.

IV. Experimental Results

In this section, we are going to experimentally demonstrate the performance of R-VIO2 using EuRoC benchmark [22] and our own sensor platform. On EuRoC dataset, we compare with VINS-Mono [23], a state-of-the-art visual-inertial navigation algorithm,[6] and our previous work, R-VIO.[7] Note that VINS-Mono enables relinearization by default, while R-VIO2 does *not* do that in order to reveal its base performance. In the real-world tests, we use large-scale trajectories to demonstrate the effectiveness of online calibration for dealing with unknown errors in spatial-temporal parameters. It is worth to point out that thanks to the square-root formulation the *single-precision* (float32) arithmetic is used by our C++ implementation of R-VIO2, that is of resource-efficiency over the double-precision (float64) ones used by the counterparts. All the tests run on a laptop with Core i7-4710MQ 2.5 GHz×8 CPU in *real time*.

A. EuRoC Dataset

This dataset contains 11 sequences which were collected by a flying quadrotor equipped with a VI-sensor (200 Hz IMU, and 20 Hz dual cameras of 752×480 pixels). Especially, only the left-camera images are used as visual inputs. For VINS-Mono, we tested its odometry pipeline, and used its default settings provided with the code for best performance. While, R-VIO and R-VIO2 used the same settings, where a sliding window of 15 relative poses was kept, with 200 Shi-Tomasi features [24] being tracked across the images. In particular, the feature tracks longer than the size of window were split into sub-tracks for use, while the sensor noise parameters directly used their raw values provided by the dataset.

We first studied the performance of estimators on the global pose (orientation and position) estimation. R-VIO was treated as baseline for R-VIO2 as it does not have online calibration (OC) function, while for VINS-Mono we turned off its online calibration using configuration file and used that as baseline. Especially, to test the convergence of online spatiotemporal calibration for VINS-Mono and R-VIO2, we initialized their corresponding parameters with the closet orthogonal rotation (for ${}^{C}_{I}\boldsymbol{\theta}$), zero translation (for ${}^{C}\mathbf{p}_I$) and zero offset (for t_d). In contrast to that, their baseline algorithms used the values provided by the dataset for those parameters. We calculated the root mean squared error (RMSE [25]) for pose estimates against ground truth of all 11 sequences using the evaluation toolbox [26],[8] and presented the results in Table I. We should note that, comparing with the baseline algorithm, both VINS-Mono and R-VIO2 achieved improved accuracy with online spatiotemporal calibration. Note also that, R-VIO2 achieved very competitive accuracy to VINS-Mono, even though the relinearization was not used. For better illustration, we depict the estimated trajectories of VINS-Mono (w. OC on) and R-VIO2 with ground truth in Fig. 1. Table I also includes the comparison of the average runtime of single-step estimation (excluding the image processing), which reveals the superior computation speed of R-VIO and R-VIO2, that is *tens* times faster than VINS-Mono. It is also impressive to find that R-VIO2 is almost twice faster than EKF-based R-VIO, for that there are two primary reasons: 1) Although R-VIO2 has more states than R-VIO, the number of entries for computation in its square root information matrix is merely *half* of that in the covariance matrix of R-VIO, thus the memory cost of R-VIO2 is lower than that of R-VIO; and 2) In-place QR-based update combined with back substitution makes the computational time complexity of R-VIO2 *linear* to the number of measurements and quadratic to the number of states. Note that, R-VIO can achieve the same linear complexity with the aid of extra QR-based model compression [1], however, the complexity of EKF update is still cubic to the number of states. Thanks to that, R-VIO2 is allowed to perform relinearization while keeping the possibility of running at full (or even higher) image rate (e.g., 10 iterations may cost ∼30 milliseconds which

[6]https://github.com/HKUST-Aerial-Robotics/VINS-Mono

[7]https://github.com/rpng/R-VIO

[8]https://github.com/uzh-rpg/rpg_trajectory_evaluation (`posyaw` alignment)

TABLE I
COMPARISON OF ABSOLUTE POSE ACCURACY (RMSE) AND AVERAGE TIME COST ON EUROC DATASET

		VINS-Mono (OC Off)			R-VIO			VINS-Mono (OC On)			R-VIO2		
	Length	Ori.	Pos.	Time	Ori.	Pos.	Time	Ori.	Pos.	Time	Ori.	Pos.	Time
	[m]	[rad]	[m]	[ms]	[rad]	[m]	[ms]	[rad]	[m]	[ms]	[rad]	[m]	[ms]
V1_01_easy	58.6	0.111	0.089	69	0.107	0.081	6	**0.097**	**0.079**	76	0.101	0.083	**3**
V1_02_medium	75.9	0.057	0.111	50	0.033	0.108	6	**0.039**	**0.086**	57	0.041	0.091	**3**
V1_03_difficult	79.0	0.107	0.189	35	0.022	0.121	5	0.088	0.201	42	**0.040**	**0.089**	**2**
V2_01_easy	36.5	0.037	0.088	35	0.039	0.158	5	**0.028**	**0.071**	69	0.040	0.117	**3**
V2_02_medium	83.2	0.074	0.160	47	0.031	0.163	5	0.044	0.123	55	**0.034**	**0.114**	**3**
V2_03_difficult	86.1	0.056	0.278	27	0.078	0.275	5	0.061	0.250	35	**0.021**	**0.116**	**2**
MH_01_easy	80.6	0.013	0.175	67	0.037	0.187	5	**0.017**	**0.150**	70	0.045	0.175	**5**
MH_02_easy	73.5	0.015	0.221	64	0.021	0.305	5	0.014	0.181	71	**0.021**	**0.167**	**5**
MH_03_medium	130.9	0.033	0.221	65	0.022	0.228	5	0.020	0.189	70	**0.018**	**0.185**	**4**
MH_04_difficult	91.7	0.014	0.371	58	0.033	0.284	6	0.018	0.285	67	**0.019**	**0.248**	**3**
MH_05_difficult	97.6	0.010	0.352	61	0.024	0.421	6	**0.016**	**0.305**	65	0.017	0.318	**3**
Mean Value	–	0.048	0.205	53	0.041	0.212	5	0.040	0.175	62	**0.036**	**0.155**	**3**

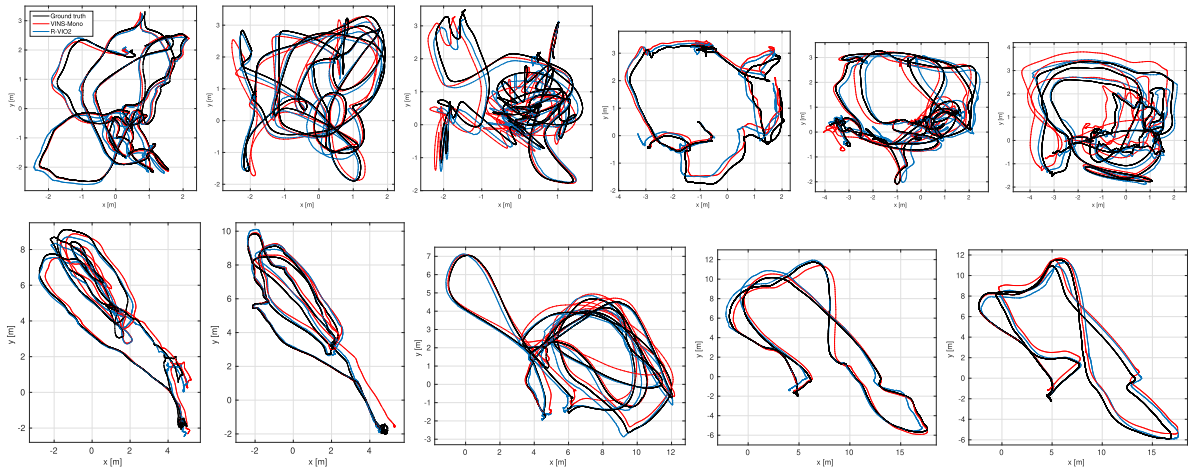

Fig. 1. Estimated trajectories on EuRoC dataset: R-VIO2 (blue), VINS-Mono (red), and ground truth (black).

TABLE II
COMPARISON OF ONLINE CAMERA-IMU SPATIOTEMPORAL CALIBRATION PRECISION ON EUROC DATASET

	VINS-Mono			R-VIO2		
	$\Delta^C_I\boldsymbol{\theta}$	$\Delta^C\mathbf{p}_I$	Δt_d	$\Delta^C_I\boldsymbol{\theta}$	$\Delta^C\mathbf{p}_I$	Δt_d
	[x: deg, y: deg, z: deg]	[x: mm, y: mm, z: mm]	[ms]	[x: deg, y: deg, z: deg]	[x: mm, y: mm, z: mm]	[ms]
V1_01_easy	[+0.15, –0.10, –0.03]	[+9.5, +4.3, –10.9]	+0.3	[–0.11, –0.17, –0.03]	[+6.2, –10.2, –10.6]	–3.5
V1_02_medium	[+0.15, –0.15, –0.02]	[+11.3, –0.4, –3.3]	+0.1	[–0.05, –0.01, –0.07]	[–5.3, –13.0, –9.9]	–5.5
V1_03_difficult	[+0.17, –0.13, –0.02]	[+13.0, +1.5, –0.3]	+0.2	[–0.17, –0.23, –0.00]	[+1.0, –12.9, +2.8]	–1.2
V2_01_easy	[+0.14, +0.03, –0.03]	[+10.7, –6.3, –10.4]	+0.3	[+0.00, –0.07, –0.03]	[–8.3, –10.9, –8.3]	–1.0
V2_02_medium	[+0.24, –0.00, –0.02]	[+8.8, –0.1, –0.5]	+0.2	[–0.04, –0.25, –0.02]	[–3.5, –10.3, +0.0]	–1.0
V2_03_difficult	[+0.26, +0.02, +0.04]	[+1.7, –3.0, +0.6]	+0.3	[+0.04, –0.34, +0.07]	[+0.7, –9.9, –3.5]	–2.6
MH_01_easy	[+0.06, –0.09, –0.02]	[+6.7, –6.3, –6.6]	+0.1	[–0.08, –0.03, +0.00]	[–6.9, –8.5, –5.8]	–4.6
MH_02_easy	[+0.06, –0.06, –0.07]	[+12.4, +5.8, –7.7]	+0.1	[–0.11, –0.18, –0.10]	[+9.5, –12.4, +28.2]	+0.2
MH_03_medium	[+0.18, –0.11, –0.03]	[+9.1, –0.8, –9.9]	+0.3	[–0.11, –0.25, –0.08]	[+1.4, –13.6, –5.7]	–1.0
MH_04_difficult	[+0.13, –0.15, +0.01]	[+19.1, +8.8, –30.2]	+0.0	[–0.20, –0.17, +0.03]	[+0.0, –10.5, +2.2]	–0.3
MH_05_difficult	[+0.06, –0.16, +0.00]	[+9.7, +0.5, –10.4]	+0.3	[–0.20, –0.02, +0.06]	[–0.3, –14.4, +13.5]	+0.4

is less than the interval between two images), and will be one direction of our future research. Such advantage is important to the resource-constrained applications, such as the navigation on drones and the AR/VR on smartphones, where the saved time can be used for image processing, path planning, or scene rendering.

Moreover, we compared the results of online calibration of VINS-Mono and R-VIO2 with the fiducial value provided by the dataset that came from an offline batch optimization [22]. As the camera and IMU were synchronized by hardware, the expected value of t_d was assumed close to zero. The estimation differences were summarized in Table II where we used small-scale units to fit their magnitudes. For $\Delta^C_I\boldsymbol{\theta}$, we first computed error rotation matrix, $\Delta^C_I\mathbf{R}$, and then converted it into Euler angles. Note that, the spatial calibrations of VINS-Mono and R-VIO2 reached the same precision, however, the differences between the temporal calibration results here, in our opinion, stem from the different measures used for t_d. For R-VIO2, we relate t_d to the changes of the IMU (or camera) poses in 3D space. While, for VINS-Mono, t_d was related to the changes of the feature locations in 2D image plane [23]. Nevertheless, comparing with the IMU sampling interval ($\sim$5 milliseconds) our results of t_d (Δt_d) are fairly consistent with the reality.

B. Real-World Experiment

We further performed tests using our own sensor platform that included a Microstrain 3DM-GX3-35 IMU (500 Hz) and

Fig. 2. Our sensor platform (w/ the images of environments).

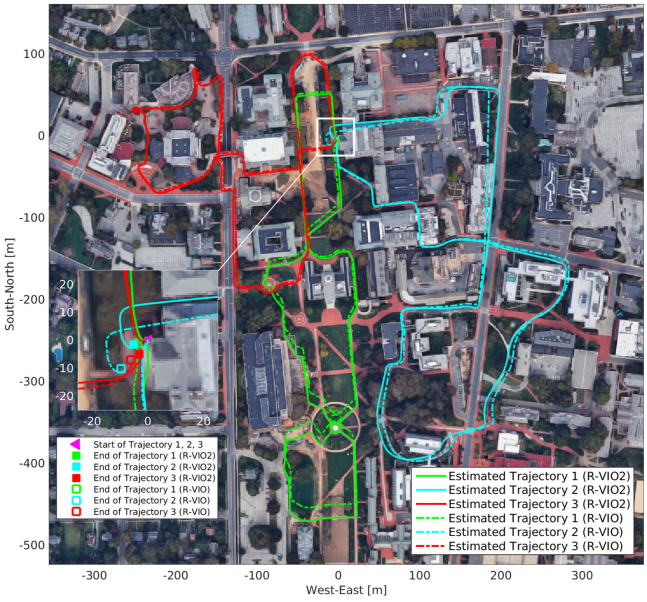

Fig. 3. Estimated trajectories over the map of UD campus.

one Point Grey Chameleon camera of 644×482 pixels (20 Hz) (see Fig. 2). We collected data by holding this sensor rig and walking in the main campus of University of Delaware (UD) in three different days, where our trajectories covered most daily routes across the campus. Over 45-minute visual-inertial data of three trajectories, each of which was about 1.5 kilometers, was collected in real time, and the walking speed was about 1.5 meters per second. Especially, as most parts of trajectories were very close to or crossed the buildings, GPS track became not reliable. Instead, the Google map of campus was used as the ground truth regarding its high precision.

As a common practice, we calibrated our sensor platform using Kalibr[9] toolbox before collecting the data. Note that, in the tests we used the corresponding result as the initial guess for spatial and temporal parameters, because in such walking scenario the sensors did not perform 6DOF motion very often which may easily prevent convergence of online calibration, and hence cause performance degradation. However, as there may still be unknown errors in the offline calibration result, online calibration can be used to refine those parameters. As before, we let R-VIO2 track 200 features across the images, while the sliding window was enlarged to include 20 relative poses. In particular, we reran all the tests with R-VIO which used the calibration result of Kalibr as true for comparison.

Fig. 3 shows the estimated trajectories that are overlaid on the map of UD campus. All the results have been aligned with their true trajectories (e.g., lanes and sidewalks) where we collected the data, from the start point. We can find that R-VIO2 outperforms R-VIO in the localization accuracy by performing online calibration, especially when we check the coincidence of estimates with their true paths. Also, as we started and ended collecting data at the same point, the final position error (i.e., close-loop drift) equals to the estimate of the last IMU position, as summarized in Table III. A zoomed subplot is thus shown in Fig. 3, where the end points of R-VIO2 are much closer to the start point of the trajectories. We also found that there existed a 4-millisecond difference in t_d before and after the online calibration. Considering the IMU sampling interval (∼2 milliseconds), here R-VIO might have misused two IMU measurements for motion prediction at every timestep, which should be its main source of errors. However, this may be hard to be noticed in practice when the duration of test was not long enough. As a summary, the final position errors of R-VIO2 are only 0.02%, 0.35% and 0.42% of the traveling distance for three respective trajectories, and its single-step estimation only costs 4 milliseconds in average.

[9]https://github.com/ethz-asl/kalibr

TABLE III
COMPARISON OF FINAL POSITION ERRORS (W/O AND W/ ONLINE CALIBRATION) OF THE ESTIMATED TRAJECTORIES

	Length / Duration	Close-loop drift in x-, y- and z-axis [m]	
		R-VIO	R-VIO2
T1	1.48km / 15m40s	[0.98, -3.70, -3.09]	[**0.19**, **0.18**, **-0.21**]
T2	1.56km / 16m43s	[8.77, -10.94, -3.99]	[**0.58**, **-5.43**, **-0.39**]
T3	1.39km / 15m24s	[5.75, -7.48, -3.11]	[**4.23**, **-4.00**, **0.52**]

V. CONCLUSION AND FUTURE WORK

In this paper, we have developed a new efficient approach for realizing robocentric visual-inertial odometry which offers consistent observability properties for probabilistic state estimator. Our robocentric models, including IMU motion model and camera measurement model, are applied for square-root information-based state estimation. Especially, the visual and inertial cost terms, as well as the a priori term are derived for the corresponding MAP minimization problem. We also take into account the influence of unknown parameter errors in the relative spatial configuration and timing between sensors, and compensate for it by performing spatiotemporal calibration online. Extensive results of public benchmark dataset and our large-scale real-world experiment demonstrate the competitive accuracy and superior time efficiency of our novel algorithm, R-VIO2, against the state-of-the-art counterparts. As a benefit of computational efficiency, we are going to enable iteration in R-VIO2 to reduce linearization errors and improve its accuracy. We will further deploy R-VIO2 on the mobile platforms, such as drones, smartphones and AR glasses, to investigate its application in different resource-constrained scenarios.

APPENDIX

1) Jacobians W.r.t. $\mathbf{x}_{\mathcal{W}}$ and $\boldsymbol{\lambda}$ (See (31)): By treating $\mathbf{x}_{\mathcal{P}}$ as constant and following the chain rule, we have[10]

$$\mathbf{H}_{i,\boldsymbol{\lambda}} = \mathbf{H}_{i,\text{proj}} \frac{\partial \mathbf{h}_i}{\partial \tilde{\boldsymbol{\lambda}}}, \quad \mathbf{H}_{i,\mathcal{W}} = \mathbf{H}_{i,\text{proj}} \frac{\partial \mathbf{h}_i}{\partial \tilde{\mathbf{x}}_{\mathcal{W}}} \tag{51}$$

$$\mathbf{H}_{i,\text{proj}} = \hat{h}_{i,3}^{-1} \begin{bmatrix} 1 & 0 & -\hat{h}_{i,1}\hat{h}_{i,3}^{-1} \\ 0 & 1 & -\hat{h}_{i,2}\hat{h}_{i,3}^{-1} \end{bmatrix} \tag{52}$$

[10]$\lfloor \cdot \rfloor$ is a 3×3 skew-symmetric matrix derived from a 3×1 vector [19].

$$\frac{\partial \mathbf{h}_i}{\partial \tilde{\boldsymbol{\lambda}}} = \begin{bmatrix} \frac{\partial \mathbf{h}_i}{\partial \tilde{\mathbf{u}}} \frac{\partial \tilde{\mathbf{u}}}{\partial [\tilde{\phi}, \tilde{\psi}]^\top} & \frac{\partial \mathbf{h}_i}{\partial \tilde{\rho}} \end{bmatrix}$$
$$= \begin{bmatrix} {}^{i}_{1}\hat{\mathbf{C}} \begin{bmatrix} -\sin\hat{\phi}\sin\hat{\psi} & \cos\hat{\phi}\cos\hat{\psi} \\ \cos\hat{\phi} & 0 \\ -\sin\hat{\phi}\cos\hat{\psi} & -\cos\hat{\phi}\sin\hat{\psi} \end{bmatrix} {}^{C_i}\hat{\mathbf{p}}_{C_1} \end{bmatrix} \quad (53)$$

$$\frac{\partial \mathbf{h}_i}{\partial \tilde{\mathbf{x}}_{\mathcal{W}}} = \begin{bmatrix} \frac{\partial \mathbf{h}_i}{\partial \delta^2_1 \boldsymbol{\theta}} & \frac{\partial \mathbf{h}_i}{\partial {}^{R_1}\tilde{\mathbf{p}}_{R_2}} & \cdots & \frac{\partial \mathbf{h}_i}{\partial \delta^{N+1}_N \boldsymbol{\theta}} & \frac{\partial \mathbf{h}_i}{\partial {}^{R_N}\tilde{\mathbf{p}}_{R_{N+1}}} \end{bmatrix}$$

$$\frac{\partial \mathbf{h}_i}{\partial \delta^n_{n-1}\boldsymbol{\theta}} = {}^{C}_{I}\mathbf{C}_{\hat{\tilde{q}}} {}^{i}_{1}\mathbf{C}_{\hat{\tilde{q}}} \lfloor {}^{I}_{C}\mathbf{C}_{\hat{\tilde{q}}} \hat{\mathbf{u}} + \hat{\rho}({}^{I}\hat{\mathbf{p}}_C - {}^{R_1}\hat{\mathbf{p}}_{R_n}) \rfloor {}^{1}_{n}\mathbf{C}_{\hat{\tilde{q}}},$$

$$\frac{\partial \mathbf{h}_i}{\partial {}^{R_{n-1}}\tilde{\mathbf{p}}_{R_n}} = -\hat{\rho} {}^{C}_{I}\mathbf{C}_{\hat{\tilde{q}}} {}^{i}_{1}\mathbf{C}_{\hat{\tilde{q}}} {}^{1}_{n-1}\mathbf{C}_{\hat{\tilde{q}}}, \text{ for } n = 2, \ldots, i. \quad (54)$$

2) Jacobians W.r.t. $\mathbf{x}_{\mathcal{P}}$ (See (31)): As a whole, we have

$$\mathbf{H}_{i,\mathcal{P}} = \mathbf{H}_{i,\text{proj}} \begin{bmatrix} \frac{\partial \mathbf{h}_i}{\partial \delta^C_I \boldsymbol{\theta}} & \frac{\partial \mathbf{h}_i}{\partial {}^{C}\tilde{\mathbf{p}}_I} & \frac{\partial \mathbf{h}_i}{\partial \tilde{t}_d} \end{bmatrix} \quad (55)$$

Similarly, we can compute the Jacobians of spatial parameters by treating $\boldsymbol{\lambda}$ as constant, so that

$$\frac{\partial \mathbf{h}_i}{\partial \delta^C_I \boldsymbol{\theta}} = (\lfloor {}^{i}_{1}\hat{\mathbf{C}}\hat{\mathbf{u}} \rfloor - \hat{\rho} \lfloor {}^{i}_{1}\hat{\mathbf{C}} {}^{C}\hat{\mathbf{p}}_I \rfloor)(\mathbf{I}_3 - {}^{i}_{1}\hat{\mathbf{C}}) + \hat{\rho} \lfloor {}^{C}_{I}\mathbf{C}_{\hat{\tilde{q}}} {}^{R_i}\hat{\mathbf{p}}_{R_1} \rfloor,$$

$$\frac{\partial \mathbf{h}_i}{\partial {}^{C}\tilde{\mathbf{p}}_I} = \hat{\rho}(\mathbf{I}_3 - {}^{i}_{1}\hat{\mathbf{C}}) \quad (56)$$

Next, by noticing that ${}^{C_1}\mathbf{p}_L$ and $\mathbf{x}_{\mathcal{W}}$ are correlated with t_d, we compute the Jacobian of t_d following the chain rule:

$$\frac{\partial \mathbf{h}_i}{\partial \tilde{t}_d} = \frac{\partial \mathbf{h}_i}{\partial \tilde{\mathbf{u}}} \frac{\partial \tilde{\mathbf{u}}}{\partial \tilde{t}_d} + \frac{\partial \mathbf{h}_i}{\partial \tilde{\rho}} \frac{\partial \tilde{\rho}}{\partial \tilde{t}_d} + \frac{\partial \mathbf{h}_i}{\partial \tilde{\mathbf{x}}_{\mathcal{W}}} \frac{\partial \tilde{\mathbf{x}}_{\mathcal{W}}}{\partial \tilde{t}_d} \quad (57)$$

$$\frac{\partial \mathbf{h}_i}{\partial \tilde{\mathbf{u}}} = {}^{i}_{1}\hat{\mathbf{C}}, \quad \frac{\partial \tilde{\mathbf{u}}}{\partial \tilde{t}_d} = {}^{C_1}\hat{\boldsymbol{\omega}} \times \hat{\mathbf{u}} \quad (58)$$

$$\frac{\partial \mathbf{h}_i}{\partial \tilde{\rho}} = {}^{C_i}\hat{\mathbf{p}}_{C_1}, \quad \frac{\partial \tilde{\rho}}{\partial \tilde{t}_d} = {}^{C_1}\hat{\mathbf{v}} \cdot \hat{\mathbf{u}} \quad (59)$$

$$\frac{\partial \tilde{\mathbf{x}}_{\mathcal{W}}}{\partial \tilde{t}_d} = \begin{bmatrix} {}^{I_2}\hat{\boldsymbol{\omega}}^\top & {}^{I_1}\hat{\mathbf{v}}^\top_{I_2} & \cdots & {}^{I_{N+1}}\hat{\boldsymbol{\omega}}^\top & {}^{I_N}\hat{\mathbf{v}}^\top_{I_{N+1}} \end{bmatrix}^\top \quad (60)$$

where ${}^{C_1}\hat{\boldsymbol{\omega}} = {}^{C}_{I}\mathbf{C}_{\hat{\tilde{q}}} {}^{I_1}\hat{\boldsymbol{\omega}}$ and ${}^{C_1}\hat{\mathbf{v}} = {}^{C}_{I}\mathbf{C}_{\hat{\tilde{q}}} {}^{I_1}\hat{\mathbf{v}}$ represent the rotational and translational velocities of $\{C_1\}$, respectively, and ${}^{I_{n-1}}\hat{\mathbf{v}}_{I_n} = {}^{n}_{n-1}\mathbf{C}^\top_{\hat{\tilde{q}}} {}^{I_n}\hat{\mathbf{v}}$, for $n = 2, \ldots, N+1$.

3) Proof of Lemma 1: Given the covariance matrices, $\boldsymbol{\Omega}$ and $\boldsymbol{\Lambda}$, of $\mathbf{x}$ and $\boldsymbol{\chi}$, respectively, $\boldsymbol{\Omega} = \mathbf{V}\boldsymbol{\Lambda}\mathbf{V}^\top$. With $\boldsymbol{\Omega} = (\mathbf{R}^\top\mathbf{R})^{-1}$ and $\boldsymbol{\Lambda} = (\boldsymbol{R}^\top\boldsymbol{R})^{-1} = \boldsymbol{R}^{-1}\boldsymbol{R}^{-\top}$, we have

$$\boldsymbol{\Omega} = \mathbf{V}\boldsymbol{R}^{-1}\boldsymbol{R}^{-\top}\mathbf{V}^\top = (\boldsymbol{R}\mathbf{V}^{-1})^{-1}(\boldsymbol{R}\mathbf{V}^{-1})^{-\top}$$
$$= [(\boldsymbol{R}\mathbf{V}^{-1})^\top(\boldsymbol{R}\mathbf{V}^{-1})]^{-1} = (\mathbf{R}^\top\mathbf{R})^{-1} \quad (61)$$

where $\mathbf{V}$ is invertible because it is a full-rank square matrix. Thus, the conclusion of Lemma 1 is immediate.

4) Proof of Lemma 2: Under the partition for $\mathbf{R}$ and $\mathbf{r}$:

$$\mathcal{C} = \mathcal{C}_1(\tilde{\mathbf{x}}_m, \tilde{\mathbf{x}}_r) + \mathcal{C}_2(\tilde{\mathbf{x}}_r)$$
$$= \|\mathbf{R}_{mm}\tilde{\mathbf{x}}_m + \mathbf{R}_{mr}\tilde{\mathbf{x}}_r - \mathbf{r}_m\|^2 + \|\mathbf{R}_{rr}\tilde{\mathbf{x}}_r - \mathbf{r}_r\|^2 \quad (62)$$

For $\tilde{\mathbf{x}}^{\oplus}$, in the sense of back substitution, we will first solve $\tilde{\mathbf{x}}^{\oplus}_r$ by minimizing $\mathcal{C}_2(\tilde{\mathbf{x}}_r)$, and then solve $\tilde{\mathbf{x}}^{\oplus}_m$ by minimizing $\mathcal{C}_1(\tilde{\mathbf{x}}_m, \tilde{\mathbf{x}}^{\oplus}_r)$. This means $\mathcal{C}_2 = \|\mathbf{R}_{rr}\tilde{\mathbf{x}}_r - \mathbf{r}_r\|^2$ includes *all* the information for solving $\tilde{\mathbf{x}}^{\oplus}_r$, which concludes Lemma 2.

REFERENCES

[1] Z. Huai and G. Huang, "Robocentric visual-inertial odometry," *Int. J. Robot. Res.*, pp. 1–23, 2019. [Online]. Available: https://journals.sagepub.com/doi/10.1177/0278364919853361

[2] G. Huang, "Visual-inertial navigation: A concise review," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2019, pp. 9572–9582.

[3] G. P. Huang, A. I. Mourikis, and S. I. Roumeliotis, "Observability-based rules for designing consistent EKF SLAM estimators," *Int. J. Robot. Res.*, vol. 29, no. 5, pp. 502–528, 2010.

[4] J. A. Hesch, D. G. Kottas, S. L. Bowman, and S. I. Roumeliotis, "Camera-imu-based localization: Observability analysis and consistency improvement," *Int. J. Robot. Res.*, vol. 33, no. 1, pp. 182–201, 2014.

[5] M. Li and A. I. Mourikis, "High-precision, consistent EKF-based visual-inertial odometry," *Int. J. Robot. Res.*, vol. 32, no. 6, pp. 690–711, 2013.

[6] 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.*, vol. 30, no. 1, pp. 158–176, Feb. 2014.

[7] G. Huang, M. Kaess, and J. J. Leonard, "Towards consistent visual-inertial navigation," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2014, pp. 4926–4933.

[8] T. Zhang, K. Wu, J. Song, S. Huang, and G. Dissanayake, "Convergence and consistency analysis for a 3-D invariant-EKF SLAM," *IEEE Robot. Automat. Lett.*, vol. 2, no. 2, pp. 733–740, Apr. 2017.

[9] F. Dellaert and M. Kaess, "Square root SAM: Simultaneous localization and mapping via square root information smoothing," *Int. J. Robot. Res.*, vol. 25, no. 12, pp. 1181–1203, 2006.

[10] M. Kaess, A. Ranganathan, and F. Dellaert, "iSAM: Incremental smoothing and mapping," *IEEE Trans. Robot.*, vol. 24, no. 6, pp. 1365–1378, Dec. 2008.

[11] K. Wu, A. Ahmed, G. A. Georgiou, and S. I. Roumeliotis, "A square root inverse filter for efficient vision-aided inertial navigation on mobile devices," in *Proc. Robot. Sci. Syst.*, 2015.

[12] K. J. Wu, C. X. Guo, G. Georgiou, and S. I. Roumeliotis, "VINS on wheels," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2017, pp. 5155–5162.

[13] P. Furgale, J. Rehder, and R. Siegwart, "Unified temporal and spatial calibration for multi-sensor systems," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2013, pp. 1280–1286.

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

[15] M. Li and A. I. Mourikis, "Online temporal calibration for camera–IMU systems: Theory and algorithms," *Int. J. Robot. Res.*, vol. 33, no. 7, pp. 947–964, 2014.

[16] G. H. Golub and C. F. Van Loan, *Matrix Computations*, vol. 3, Baltimore, MD, USA: JHU Press, 2012.

[17] W. G. Breckenridge, "Quaternions proposed standard conventions," *Jet Propulsion Laboratory, Pasadena, CA, IOM 343-79-1199*, 1979.

[18] Z. Huai and G. Huang, "Robocentric visual-inertial odometry," Univ. Delaware, Newark, DE, USA, Tech. Rep. RPNG-2018-RVIO, 2018. [Online]. Available: https://udel.edu/∼ghuang/papers/tr_rvio_ijrr.pdf

[19] N. Trawny and S. I. Roumeliotis, "Indirect Kalman filter for 3D attitude estimation," MARS, Univ. Minnesota, Minneapolis, MN, USA, Tech. Rep. 2005-002, 2005.

[20] J. Civera, A. J. Davison, and J. M. M. Montiel, "Inverse depth parametrization for monocular SLAM," *IEEE Trans. Robot.*, vol. 24, no. 5, pp. 932–945, Oct. 2008.

[21] B. Triggs, P. F. McLauchlan, R. I. Hartley, and A. W. Fitzgibbon, "Bundle adjustment–a modern synthesis," in *Proc. Int. Workshop on Vis. Algorithms*. Berlin, Germany: Springer, 1999, pp. 298–372.

[22] M. Burri et al., "The euroc micro aerial vehicle datasets," *Int. J. Robot. Res.*, vol. 35, no. 10, pp. 1157–1163, 2016.

[23] T. Qin and S. Shen, "Online temporal calibration for monocular visual-inertial systems," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2018, pp. 3662–3669.

[24] J. Shi and C. Tomasi, "Good features to track," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 1994, pp. 593–600.

[25] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, *Estimation With Applications to Tracking and Navigation*. Hoboken, NJ, USA: Wiley, 2004.

[26] Z. Zhang and D. Scaramuzza, "A tutorial on quantitative trajectory evaluation for visual (-inertial) odometry," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2018, pp. 7244–7251.